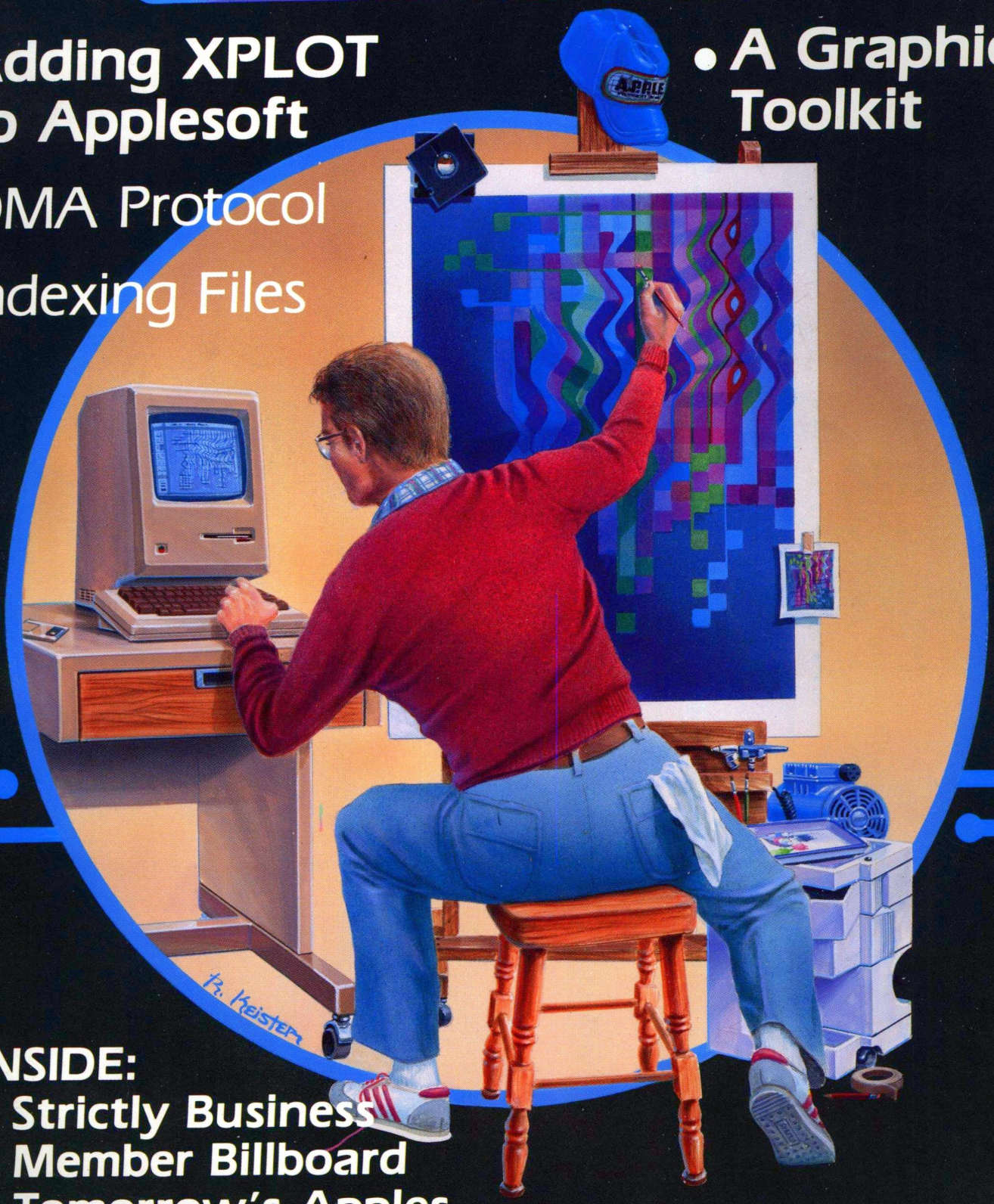
Call-A.P.P.L.E.

April 1984
Vol. 7, No. 4

Worlds Largest Apple User Group

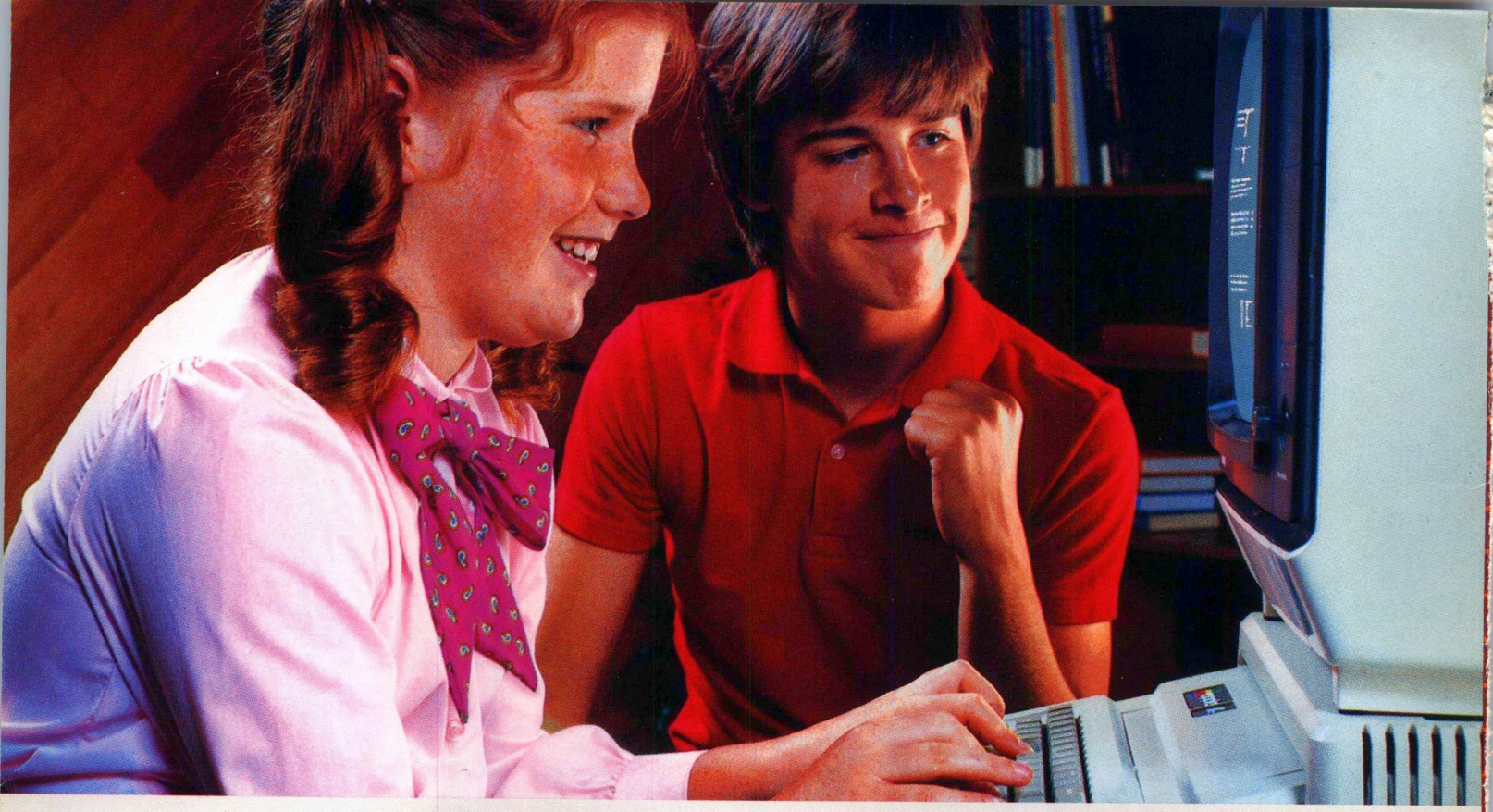
$3.00
$3.75 Canada
£2.00 UK

- Adding XPLOT to Applesoft
- DMA Protocol
- Indexing Files
- A Graphics Toolkit



INSIDE:
Strictly Business
Member Billboard
Tomorrow's Apples

Apple PugetSound Program Library Exchange



GIVE YOUR CHILDREN A GREAT TEACHER FOR THEIR APPLE.

Bring Basic Math, Algebra, Physics, Computer Literacy, Computer Concepts and Foreign Languages to your Apple II Plus or IIe

PLATO® Educational Courseware is a great teacher because it's both serious and fun. This makes it a truly exciting home learning experience... one that your children will use again and again.

PLATO makes a real adventure of topics that children often find dull... like Algebra. Difficult concepts such as roots, polynomials and quadratic equations become almost fun and therefore easier to learn.

With PLATO courseware, children study by competing against themselves and by interacting with stimulating graphics that keep them motivated. They see their progress, and experience the rewards of real achievement.

Widen your child's world

Select from a growing library of courseware for Apple II Plus and Apple IIe microcomputers. Ask about lessons also available for the TI 99/4A and Atari 800.

Ask for PLATO courseware wherever quality computer products are sold.

Discover the excitement PLATO courseware can bring to learning... in your home.

For a free catalog describing all the PLATO courseware for microcomputers: call toll-free 800/233-3784. (In California, call 800/233-3785.) Or write Control Data Publishing Co., P.O. Box 261127, San Diego, CA 92126.

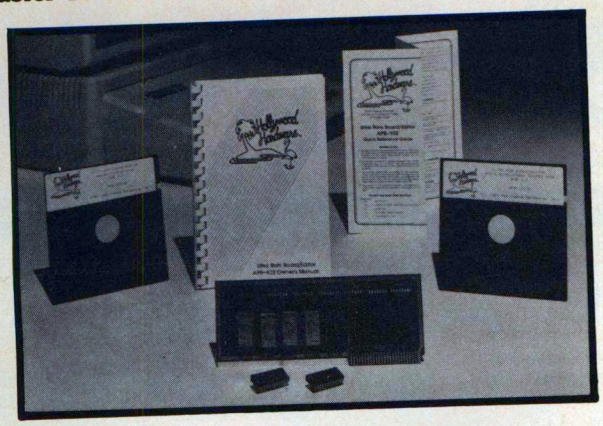
Warranty available free from Control Data Publishing Co., 4455 Eastgate Mall, San Diego, CA 92121.

PLATO
COMPUTER-BASED EDUCATION

CONTROL DATA PUBLISHING

Circle 41 on Reader Service Card

Would You Like to Make Programming Just Plain **Easier?** Easier to Learn— Easier To Do— and 3 Times **Faster?!**



The Hollywood Hardware Firmware Enhancement System

Why You Need It:

We take all the enhancements that transform the Apple from a novelty to a powerful instrument, and hook them into the operating system, installed on their own FIRMWARE card. NO disk loading, NO loss of available memory space, NO interface with other programs. The system never need be removed - it is unhooked with two keystrokes, rehooked with four. ALWAYS IN THE MACHINE, NEVER IN THE WAY.

The Response: We Get a Lot of Nice Comments:

"...Excellent product, flawless.." Phil Daley

"...Hollywood Hardware has developed an elegant solution...it makes editing an Applesoft program all it should have been in the first place, and then some."... SOFTALK review

"...The best thing for the Apple since the disk" Edward Decker

The Basic System Includes: **$189.00**

a) **The ULTRA-ROM BOARD (APB102A)** Installs 32K of firmware space in any slot of your II, II+, //e. Requires no program memory.
"Nice looking, high quality hardware."... Robert Zedelis

b) **GPLE 4.2:** Enhanced version of the original editor that ALLOWS good programming practice. GLOBAL SEARCH; automatic EDIT or REPLACE; INSERT, DELETE, FIND, ZAP; CASE CONVERT and RESTORE variables, or characters. We guarantee you will never want to be without it.
FUNCTIONS: Sits between the keyboard and the rest of the Apple - so any key can double as a FUNCTION KEY that generates a whole command string. Over 60 of the most used are provided as MACROs. System users never type "CATALOG, D1", or SAVE <filename>". Just type TWO keys: ESC 1, or ESC CTRL-S.
CUSTOMIZE and save your own macros!
"The most powerful program tool I have..." Robert Wilson
"If you program, and haven't used a line editor, get one right away..." Micro Magazine review

c) **FMS: The FIRMWARE MANAGEMENT SYSTEM** Finds and enables the desired routine with automatic bank switching. New ROMS are recognized automatically.
"...The integrated memory managment system is one of the best features of the board." Mark Simonson, Beagle Brothers.

d) **APU-1:** Over 25 UTILITIES. Invoked by the "&" key: **Ultra fast searches, Garbaged program recovery**. Instant numerical **conversions**, Graphics state controls. Language extensions like PRINT USING and IF, THEN, ELSE are included, and provided in disk form for portability.
"...P.S. love it! Can hardly wait for more!" Paul Fox

Optional Additions Include:

e) **NEW!! APU-2: supeRENUMBER, multiHIDE, multiMERGE, VARIABLE CROSS REFERENCE, AUTONUMBER.** NEW programs by Paul Johnson. Far more powerful and mistake proof than other versions...and you don't destroy your own program loading them in from disk. They are always at hand. **$35.00**

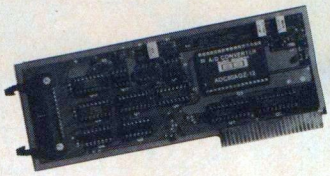
f) **NEW!! COPY ZAP by Glen Bredon - author or BIG MAC.** SMART DISK COPIER, and ZAP utilities written for our system. Also copies 40 track disks! Hit "&COPY" and watch! **$39.95**

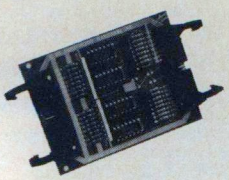
g) **NEW!! ROM DEVELOPEMENT SYSTEM:** Program your own firmware! Includes software, documentation and emulation RAM. **$99.00**

NEW NEW NEW

12 Bit 16 Channel Data Acquistion

12 BIT 16 CH. A/D SIGNAL CONDITIONING





AD16B $299.95

- ★ IRQ or NMI INTERRUPT
- ★ PRECISION SAMPLE & HOLD
- ★ EXTERNAL START CONVERT
- ★ HIGH SPEED - 25,000 CONV/SEC
- ★ 7 VOLTAGE RANGES
- ★ PRECISION REFERENCE OUT
- ★ SUCCESSIVE APPROX. CONVERTOR
- ★ COMPLETE SOFTWARE INCL. DEMOSCOPE

A16G $79.95

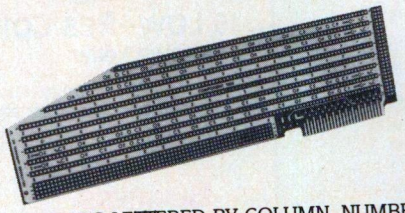
- ★ BOLT-ON FITS IN APPLE
- ★ 16 OP AMPS
- ★ EACH CHANNEL SEPARATE GAIN & FILTERING

A8D from $149.95

- ★ 2 to 8 CHANNELS
- ★ TRUE DIFFERENTIAL INPUTS
- ★ BOLT-ON FITS IN APPLE
- ★ GAIN JUMPER EACH CHANNEL
- ★ GAINS OF 1, 10, 100, 1000

EXPANDED NEW FEATURES

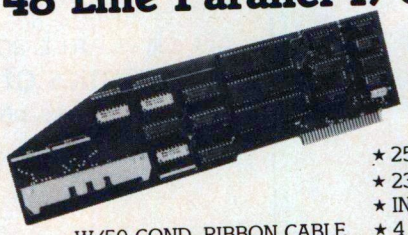
PRO-1 PROTOTYPING BOARD



THE BIGGEST THE BEST $29.95

- ★ ALL IC'S LETTERED BY COLUMN, NUMBERED BY ROW
- ★ ACCEPTS ALL STANDARD SOCKETS - UP TO 52 (16 PIN)
- ★ NUMBERED INPUT & OUPUT PINS ACCEPT I.D.S. CONNECTOR
- ★ ±5 AND ±12 SUPPLIES HANDY & WITH SPACE FOR FILTERING
- ★ GOLD PLATED EDGE CONNECTOR, PREMIUM BLACK BOARD

48 Line Parallel I/O PD48



$249.95

W/50 COND. RIBBON CABLE

- ★ 25 FILTERED LINES IN
- ★ 23 BUFFERED LINES OUT
- ★ INTERRUPT INPUT
- ★ 4 TIMERS

ADDITIONAL FINE PRODUCTS:

★ **MACROTECH MEMORY EXPANSION**	
DISCULATOR 64K	— $379.95
DISCULATOR 128K	— 459.95
★ **ANCHOR AUTOMATION MODEMS**	
VM-1 VOLKSMODEM 300 BAUD	— $64.95
VM-1 CABLE	— 9.95
SIGNALMAN MARK IL 300 BAUD	— 79.95
MARK VII, 300 BAUD, AUTO	— 109.95
MARK XII SMART MODEM	— 299.95
300/1200 BAUD AUTO ANSWER BELL 212, RS 232, HAYES* COMPATABLE	
★ **HOLLYWOOD HARDWARE T-SHIRT**	
SM, MED, LG, XLG	— 12.00

*TRADEMARK HAYES MICROCOMPUTER PRODUCTS

Hollywood Hardware 6842 VALJEAN AVENUE, VAN NUYS, CA 91406 (818) 989-1204

VISA MasterCard

*GPLE © 1983 NEIL KONZEN, SOLD UNDER LICENSE FROM SYNERGISTIC SOFTWARE / APPLE IS A REGISTERED TRADEMARK OF APPLE COMPUTERS INC.

Apple PugetSound
Program Library Exchange
21246 - 68th Ave. S.
Kent, WA 98032 - (206) 872-2245
Hours: 9am to 4pm PST, Mon-Fri.

Officers and Directors
Dick Hubert President
Bob Huelsdonk Vice-President
Dave Lingwood Secretary
Bob Clardy Treasurer
Dave deGroot Director
Val J. Golding Director
Don Williams Director

Administration
Dick Hubert Executive Director
Don Steele Asst. to Executive Director
Dave O'Leary Controller
Sally Davies Manager of Purchasing
Steve Johnson Technical Staff
Charles Stillman Manager of Education Services

Departments
Order desk
Tracing
Member Information
Education, Classes
Retail store
Dealer services
Marketing
Technical support
Testing and Evaluation
Operations

Publications
Gary Kruger Executive Publisher
Val J. Golding Editor
Kathryn Hallgrimson Suther Associate Editor
Ed Adams Art Director
Ron Keister Cover Artist
Paul Sleeper Pascal Editor
Grawin Publications Advertising
John Elliott Advertising Coordinator
Jonathan Post Contributing Editor

Staff writers
Cornelis Bongers
Wes Felty
Jim Hockenhull
Clifton M. Howard, M.D.
Ken Kashmarek
Paul Mosher
Bill Parker
Lee Reynolds
Clay Ruth
Michael Shadick
David G. Sparks
Rick Sutcliffe
Ralph H. Swerdlow, M.D.
Lee Webster

Member Services

24 hour toll-free order line:
1-800-426-3667
(except Washington state)
Orders ONLY
Information not available

Programming/Technical Assistance
Call -A.P.P.L.E. HOTLINE
(206) 872-9004
9am to 3pm - 6pm to 10pm
(Pacific time)
7 days per week

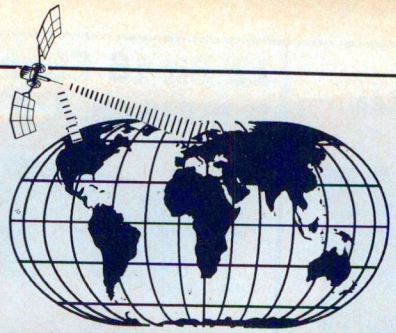
Consultants
see listing on tear-off card inside

A.P.P.L.E. Crate Bulletin Board
(206) 872-6789

Meetings
7:30pm third Tuesday of each month
A.P.P.L.E. Office
21246 68th Ave. S., Kent
(except as noted)

Membership Information
is available on postpaid tear-off card inside

A.P.P.L.E. on the SOURCE
A special section of The Source is open only to A.P.P.L.E. members who may download Call -A.P.P.L.E. programs. To access, type: CALL -APPLE from Source command mode. To register, send mail to ST1570 with your name, membership number, expiration date and Source account number.



Call-

Graphics

10 **A Graphics Tool Kit**
randi rost

part 1 introduces our new series with a description of screen mapping, then offers some assembly subroutines

17 **Adding XPLOT to Applesoft**
mark harris

as easy as pie (charts) and 179 bytes of assembly code

19 **Low-Res Color Change**
stowe keller

This short utility lets you change one plotted color to any other at the flick of a key

Business

THE SPREADSHEET SHEET:
21 **Data Analysis (Part 5)**
jeffrey jacques/phd

exploring essential concepts of the linear equation

A Call -A.P.P.L.E. BOOK REVIEW:
25 **Dynamics of VisiCalc**
jeffrey jacques/phd

a quick look at a new visi-book with feedback by the authors

27 **Strictly Business**
mike regeimbal

words of wisdom about organization structures

Telecommunications

31 **ABC's of ABBS's**
philip chien

part two describes writing or personalizing your own bulletin board, and includes a glossary of bbs terms

Hardware

FROM THE VERY CORE OF APPLE:
35 **DMA Protocol**
peter baum/apple computer/inc

a technical paper for doing direct memory access with peripheral cards

Technical

A Call -A.P.P.L.E. TECHNIQUE:
39 **Setting I/O Hooks in ProDOS**
cecil fretwell

the idea is still the same, but the method has changed for the better

A Call -A.P.P.L.E. UTILITY:
42 **Converting Random Files to ProDOS**
ken kashmarek

something the boys at apple didn't come up with

Tutorial

A Call -A.P.P.L.E. TUTORIAL:
44 **Indexing Files**
scott g smith

dos files can work well for you, but there are a few tricks to make them efficient

ADVERTISER INDEX

A.P.P.L.E. 9
Access Micro 39
Artemis Systems, Inc. 55
Beagle Brothers 40, 41
Bytes & Pieces 34
Central Point Software 20
Control Data ..... 2nd Cover
David — Data 29
Douglas Electronics 30
Emerald City Publishing, Inc. 47
Graysoft 56
Hollywood Hardware 1
Laing Electronics 52, 65
Many Software Systems 78
Maryland Computer Services 16
Micro-Ware Distributing 25
Nibble Magazine 76
Nibble Notch 12
Polytel 8
Quorum International 28
R H Programming 68
Roger Wagner Publishing ..... 3rd Cover

A.P.P.L.E.™

Apple PugetSound Program Library Exchange

Random Access

53 **The Northern Spy**
rick sutcliffe

interesting comments on topics from the dot matrix printer to peripheral cards

52 **A.P.P.L.E. Aids Blind**
patrick schwab

what do you do in a strange town when you've forgotten your monitor? Call –A.P.P.L.E.!

10 **Apple Announces Lisa, Mac Seminars**

if you're a developer, you may want to get in on one of these

YOU CAN BE . . .
53 **The Chairman Of the Board**
david I pontius

data processing and budget preparation in the home (with a twist)

Education

FLYING HIGH . . . WITH E–Z PILOT
58 **ZOOM! Big Time Program**
earl keyser

using labels with U statements

57 **Koala Pictures**
lorana a jinkerson

adding a graphics print routine to the koala drawing program

AN INTRODUCTION TO STRUCTURED PROGRAMMING:
64 **Pascal Low–Res in the High School**
ian jackson

turtle graphics is much too sophisticated for the beginners. The library unit LORES does the trick

New Products

75 **Tomorrow's Apples Today**
Kevin O Donohoe

five new graphics products and three for the Macintosh may tempt you for tomorrows acquisitions.

Letters

48 **Write –A.P.P.L.E.**
the readers

51 **A.P.P.L.E. Doctor**
bruce f field

this popular feature covers more on applewriter and printers, with just a dash of cp/m

72 **Writing the Wrongs**
the wrongdoers

corrections to "numbers" (august 1983), ampermanager, and supplemental information about "hi–res slide show" (december 1983) are covered this time around

Departments

5 **The Editor Bytes Back**
the editor

7 **Bytes From A.P.P.L.E.**
val j golding

5 **Boot**
craig macintosh

60 **Unclassifieds**

A.P.P.L.E. BITS & PIECES:
61 **Member Billboard**

reader feedback heads off billboard this issue, plus words on the system 38, along with an admission of guilt

59 **A Call –A.P.P.L.E. Quickie**

installing a non–flashing cursor in gpie

7 **The Moving Finger Writes . . .**
val j golding

the editor drops the first person plural for a message to his readers

Apple user groups may reprint without prior permisson any portion of the contents herein, provided author, title, and publication credits are given and a written notice of such use furnished to the editorial office. For commerical applications of programs herein, the author should be contacted c/o General Office. Call —A.P.P.L.E. is published monthly by Apple PugetSound Program Library Exchange, 21246 - 68th Ave. S., Kent, WA 98032. Subscriptions are not available but copies are mailed free as a benefit of A.P.P.L.E. membership.

Entire contents copyright © 1984 by Apple PugetSound Program Library Exchange.

Sibsource, Inc. 66
Softkey Publishing..... 36
Softronics..... 4
Software Banc, Inc. 26
Specific Software..... 50
Star Logic..... 33
Stone Ware..... 6
Telephone Software Connection.... 15
Titan Technologies..... Back Cover
VF Associates..... 24

Changes of address must be received at the General Office at least 30 days prior to effective date of move. Issues missed due to non-receipt of change of address notification must be separately reordered by the member at current cost. Change of address form may be found on this page.

I am movingand my present address is:

PRESENT:

NAME _____

ADDRESS _____

CITY _____

STATE _____ ZIP CODE _____

EFFECTIVE DATE _____

NEW ADDRESS:

NAME _____

ADDRESS _____

CITY _____

STATE _____ ZIP CODE _____

MEMBERSHIP NUMBER _____

* Please fill out and return this notice six weeks before moving.

NOW YOUR APPLE® CAN HOOK TO ANY MAINFRAME.

Apple is a registered trademark of Apple Computer, Inc.,
CP/M is a registered trademark of Digital Research, Inc.,
Dow Jones News/Retrieval is a registered trademark of
Dow Jones, Inc. The Source is a service mark of
Source Telecomputing Corporation,
CompuServe is a registered
trademark of CompuServe, Inc.

If you have an Apple II, Apple II Plus, or Apple IIe, we have good news for you. Now there are two inexpensive software programs that can turn your Apple into a much more valuable tool.

Softerm 1 connects you with information services.
Softerm 1 lets you retrieve information from services such as *The Source,*℠ *CompuServe®,* and *Dow Jones News/Retrieval®.* Plus, gives you the ability to access bulletin boards and send or receive electronic mail. Other features include user-defined keyboard macros, built-in phone book for automatic dialing, terminal mode line capture simultaneously to print or disk, copy screen to print or disk, and terminal status display.

Softerm 2 connects you with your company's computer.
This expanded version of Softerm lets you gain access to the information stored in your company's main computer from your home or office. With either version of Softerm, you can down load information into your Apple and capture it on your own disk in any format you choose—DOS, CP/M®, or Pascal. Also included with Softerm is a source program for your host computer to ensure compatibility with Softerm's file transfer capabilities.

Makes your Apple work exactly like any major terminal.
Softerm 2 provides complete emulation of these terminals: ADDS Regent 20, 25, 40, 60; ADDS Viewpoint; Data General D200; Datapoint 3601; DEC VT102, VT52; Hazeltine 1400, 1410, 1500, 1520; Hewlett-Packard 2622A; Honeywell VIP7205; VIP7801, VIP7803; IBM 3101 Model 10 and 20; Lear Siegler ADM-3A, ADM-5 and TeleVideo 910, 925, 950. And the list is growing all the time. We'll send you a User's Guide, handy reference card, and a telephone number to call if you need more assistance.

Softerm can't become obsolete.
We constantly improve Softerm. And we make those improvements available to you 24 hours a day, 7 days a week. To update your program, just dial our computer and transfer the improvements to your disk.

Get Softerm now.
It's the inexpensive way to turn your Apple into a much more valuable tool for your home or office. Softerm 1 is only $135; Softerm 2 is $195. Both are available now from your local dealer or from Softronics, Inc. For more information or to order, call (901) 683-6850.

SOFTRONICS
3639 New Getwell Road, Suite 10
Memphis, TN 38118. (901) 683-6850

The Editor Bytes Back

the editor

GRAPHICS from the very beginning have been one of the Hallmarks of the Apple][. When the Apple was first released in 1977, its graphics were second to none. Today, even in the face of newer technologies, it remains high on the list of contenders. Not surprisingly, interest remains high in Apple graphics, and with the advent of the Apple //e and its double resolution capabilities, not all has been said, not by a long shot.

In this issue **Randi Rost** introduces the first of his series entitled **A Graphics Toolkit,** which will show you in easy stages how to develop your own assembly language subroutines for fast running high-res games and other applications.

Mark Harris is back after a long absence and ponders if you can have DRAW and XDRAW, why not PLOT and XPLOT. **Adding XPLOT to Applesoft** shows you how. Low resolution graphics is by no means passe, and in our education section, **Ian Jackson** describes how how he uses it to teach Pascal to his high school students in New Zealand. See his **An Introduction to Structured Programming: Pascal Low-Res in the High School. Stowe Keller** has also come up with an interesting low-res utility called **Low-Res Color Change.** It will change any occurence of color A to color B at the drop of a hat (or at least a CALL to 768).

From Apple Computer comes **Peter Baum,** with a technical paper, complete with figures: **DMA Protocol,** the first time we have seen anything on the subject of Direct Memory Access protocol. Hardware types may want to turn here first.

One of the shortcomings of ProDOS is the lack of an ability to convert a random access file from DOS 3.3 to ProDOS. Staff writer **Ken Kashmarek** has written a BASIC program which runs under both 3.3 and ProDOS, and converts a random access 3.3 file to a sequential type, then, when ProDOS CONVERT has been run, the CONVERT.RANDOM program in Ken's **Converting Random Files to ProDOS** will convert the (now) sequential file back to its original random form.

Patiently, **Scott G. Smith** has been waiting in the wings with a tutorial on **Indexing Files,** which describes in simple terms, common techniques such as a linked list (which is the type of file used for messages on many bulletin board systems). Speaking of which, **Philip Chien** is back with us to conclude his **ABC's of ABBS's.**

Have at it!

Boot

Pick a Number Between pfs®:file and dBASE II®....

You've outgrown the simple file management program you've been using. Perhaps you need more than one diskette of data in a file, or greater math capabilities, or password file protection, or a more powerful report generator. You know which program the hot-shot programmer down the street uses, but you still think that "multiple relations" refers to your in-laws, and you're not sure you *want* to know "query language". Besides, if it's so easy to use, why all those expensive training seminars and books and programs to make it easier? Don't trade in your computer on a bunch of used file cabinets. . . .

Check out the alternatives . . .

ADVANCED DB MASTER

The most powerful user-oriented information management system available for the IBM PC®, PC/XT®, 100% compatibles, and SAGE™ II/IV computers. Works with hard disk or floppies. 256K RAM required.

- Up to 44 diskettes of data per file!
- Built-in letter writing/mail merge capability
- Exchange data with 1-2-3™, VisiCalc®, Multiplan™, etc.
- Password file protection
- Keystroke macros

Retails for $595.

Try Before You Buy!

For a working sample version of Advanced DB Master, see your Stoneware dealer, or send $9.95 (refundable with purchase) to Sample Version, Stoneware, Inc., 50 Belvedere St., San Rafael, CA 94901.

DB MASTER 4 PLUS

We integrated all the functions* of Utility Paks #1 and #2 into DB Master Version Four and created DB Master 4 Plus. Now you get more features than ever for Apple® II+/IIe or Franklin Ace™ 1000/1200 information management—**with no increase in price!**

- Up to 100 or more diskettes of data per file!
- VisiFile™ and PFS file converters
- File restructure capability
- DIF™ translators
- Global editor (including date math)
- Label printer up to 5 across)
- Damaged file recovery

*(except Transaction File Merge)

Retails for $350.

Free Brochure

For a free brochure describing DB Master 4 Plus, see your local Stoneware dealer.

Power Without Programming!

STONEWARE®

STONEWARE INCORPORATED • 50 BELVEDERE STREET • SAN RAFAEL, CA 94901
TELEPHONE (415) 454-6500 • TELEX II: 910-384-2016 STONEWARE SRFL

REGISTERED TRADEMARKS:
Stoneware (Stoneware, Incorporated) • IBM PC and IBM PC/XT (International Business Machines Corporation) • dBASE II (Ashton-Tate) • pfs (Software Publishing Corporation) • VisiCalc (VisiCorp)

TRADEMARKS:
Advanced DB Master and DB Master (DB Master Associates) • Apple II+ and IIe (Apple Computer, Inc.) • DIF (Software Arts Corp.) • Franklin ACE (Franklin Computer Corporation) • 1-2-3 (Lotus Development Corporation) • Multiplan (Microsoft Corporation) • VisiFile (VisiCorp)

Circle 1 on Reader Service Card

Bytes from A.P.P.L.E.

val j golding

COMING soon from A.P.P.L.E.: blank 3½" Macintosh diskettes! One of the things we try and do at A.P.P.L.E. in considering new products is to determine what the "in" thing is, and how "in" it will be six months or a year from now. We think the Macintosh will be around long after many of the numerous other machines are gone, so you can watch this spot next month and see how much we can save you on Mac diskettes.

One of our very popular software items has been our collection of several hundred public domain diskettes, priced at just $4.00 each, which essentially represents our cost of the media, plus production and shipping. Because of the very low price, we have not, up to this time, been able to apply our standard replacement policy to the PDS disks. Effective April 15th, which coincides with the issue date of our spring catalog, the price for the public domain disks **(As Is Software)** will be $5.00, and will include a replacement policy for defective disks. The As Is, however, retains its original meaning; we can not support the programs or guarantee that they are bug-free.

Still and all, the PDS disks are really a "best buy," particularly if you are just starting a software collection. We have available a complete list of PDS disks which is updated from time to time, so if it has been a while since you have seen the list, you might want to get a new one. Send a stamped, self-addressed, legal size envelope with .37 postage and ask for our PDS list.

Another year has more than rolled by; we are in fact a third of the way through 1984, and our 1983 bound volumes will be out of the bindery shortly, which brings us to our special of the month. Not without a catch, however. But to offer you a savings of almost 50%, we need to consolidate orders, thus to get the special, we ask that you either join as a new member, or order $50.00 worth of products at the same time.

Here's how things stack up: The member price for the 1983 bound volume will be $18.50, the four Anthology diskettes (9a, 9b, 10a, 10b) for 1983 are $12.50 each. That's a total member price of $68.50, but you can have it for only $35 plus an order of $50 or more from our catalog, or with a new membership. Remember the name, it's the **1983 Anthology Special.** The product code is SPANTH, and the overseas shipping weight is seven pounds. Just in case you've forgotten, the Anthology diskettes contain the programs published in *Call -A.P.P.L.E.* The 1983 Anthology Special will expire on May 31, 1984.

The Moving Finger . . .

Val J. Golding

IT is common practice in many magazines for the editor to avoid the use of the personal pronoun "I," for myriad reasons. It seems less self-serving, less egotistic, and most often the editor or editorial writer is expressing thoughts and ideas of a larger whole, the staff, the (in this case) club, etc.

In the case of this editor, that form has been used on only a couple of previous occasions, thus, nurtured by its absence, the use of the personal pronoun attaches a note of particular significance or importance to the message.

I am very proud to hold A.P.P.L.E. membership number one, and to have just recently been re-elected a director of an organization which I helped develop. I am constantly reminded — and amused — by my co-workers and friends, who delight in pointing out to me the statement I made in **Call -A.P.P.L.E.** Vol. I, No. 1, February 1978, "Regretfully, I do not have the time available to continue this project beyond the formative stage . . ."

Call -A.P.P.L.E. has come a long way since that typewritten, single-page newsletter. It's been fun every step of the way — a real ball — and I wouldn't have missed any of it for anything. It is interesting to reflect back on that first newsletter and re-read the goals I set for the club: ". . . establishing a software library . . . programming and technical assistance . . . the exchange of information . . . and the publication of a newsletter."

All of these original goals were met and exceeded, and A.P.P.L.E., by fits and starts has become the world's largest Apple Computer user group.

As I said at the beginning, I'm proud to have been a part, and proud of my membership card, a piece of plastic which, I assure you, will not become encased in bronze, but will continue to be carried in my wallet as a reminder of my ongoing obligations to you and A.P.P.L.E.

Even while loving one's job, one's work, one can become stagnant, and for a creative person, this is not a healthy condition.

Sometimes also, circumstances intervene in one's personal life, factors never considered or anticipated, but which, nevertheless, have their effect. Then too, there are once-in-a-lifetime opportunities; I've been blessed with two, the first when I quit my 9-5 job to become editor of **Call -A.P.P.L.E.**, and the second, which has just recently come my way.

Over these seven years, I've been the voice of the "Hotline" you've called for assistance (evenings and weekends for the last two years). It has been my pleasure to be able to talk to so many of our members and perhaps to offer solutions to their problems. Maybe even the "Dear Abby" of Apples! Moreover, A.P.P.L.E. has had booths at the majority of the Applefests and nationwide computer shows over the last few years, most of which I've been fortunate enough to attend.

I've always looked forward to each new show, even though I return with mountains of work undone, because it is at these affairs that I have the rare opportunity to shake hands and say hello to many of you who, up to that point, were voices and names. In a somewhat different perspective, I hope to be able to match up those voices and names with faces in the time ahead.

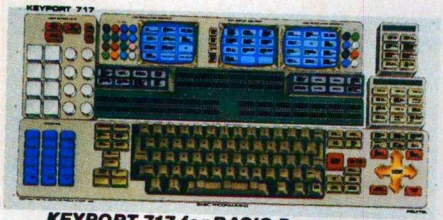
All of a sudden, as I write these words, genuine tears well up as the thought comes to mind, "This is really it," and I realize this is to be my "swan song," my last issue of **Call -A.P.P.L.E.** as editor.

continued on p. 68

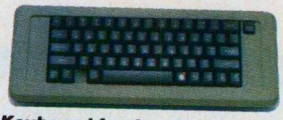
BREAK THE KEYBOARD BARRIER

KEYPORT 717®

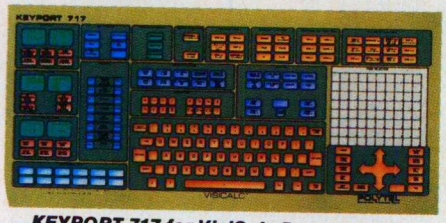
FOR APPLE II, II+ AND IIIe



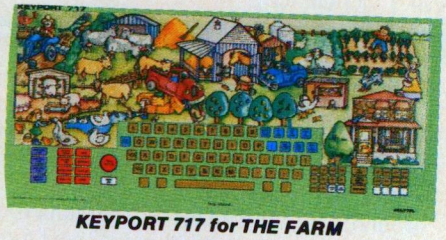
KEYPORT 717 for BASIC Programming



Keyboard for Any Application



KEYPORT 717 for VisiCalc Programming



KEYPORT 717 for THE FARM

The KEYPORT 717 is an advanced technology keyboard with 717 touch-sensitive function keys. These keys allow the commands for a specific application to be placed in front of the user in colorful graphics. Each application has its own KEYPORT overlay.

KEYPORT 717 SPECIFICATIONS

No. of keys	717
Shift keys	2
Size of overlay	11″ x 24″
Size of case	12″ x 25″ x 2″
Connector	9 Pin or 16 Pin DIP
Cable length	5 to 50 feet

The KEYPORT can assign an entire command, or string of commands, to a single key. This greatly increases user and programmer efficiency, and decreases the frequency of syntax errors. A graphic overlay designates which of the 717 keys are activated, and these keys are logically grouped and color coded to organize workflow and increase user concentration. A booter diskette is linked to the overlay, and activates the particular set of keys.

The KEYPORT 717 does not disable the standard keyboard. It is connected to the computer via the game port to allow both the keyboard and the KEYPORT to be used concurrently.

POLYTEL's initial software packages include frequently used and popular programs (BASIC Programming, VISICALC, and THE FARM) with word processing and other programs coming soon. Your **productivity** will grow while using the **affordable** KEYPORT system.

BASIC PROGRAMMING

The BASIC Programming overlay includes keys for all BASIC and DOS commands PLUS features like AUTO-LINE NUMBER, four-directional non-destructive cursor for editing, AUTO-REPEAT for all keys, user definable keys, COLOR keys, and more.

VISICALC

The VISICALC overlay includes all numeric, formatting, and text capabilities in an easy-to-use layout. Spreadsheet construction and labeling is now easier through the use of grouped and understandable single key commands. Other features include eight directional cursor control, printing macros, user definable keys, and a screen memory map for one-stroke cursor positioning.

EDUCATION

THE FARM is the beginning of an exciting educational line of software packages. This extremely colorful overlay has a farmyard scene in which your children can identify farm animals, ask questions, play games, or even write a story about the experience.

PROGRAMMERS

The KEYPORT 717 is not only super User Friendly, but also Programmer Friendly. In addition to using the packages described above, you can design your own applications and overlays around the KEYPORT. New applications can be programmed easily, without menus and syntax analysis, because every function can be associated to a unique key.

Every KEYPORT comes complete with a set of software tools and two template programmer overlays to help you custom design a KEYPORT overlay for new or existing programs. Potential applications include business, home use, education, process control, and automation.

Copyright protection for your program is enhanced by requiring the custom overlay for its use.

SYSTEM REQUIREMENTS

At present, the KEYPORT 717 is compatible with the APPLE II, II+, and IIe. A 48K memory and one disk drive is required. No interface card needed. The cable, booting diskette, programmer overlays, and BASIC Programming package are included with every KEYPORT 717.

New KEYPORT products are now being developed for IBM, Commodore, and other popular microcomputers.

TO ORDER YOUR KEYPORT 717, SEE YOUR LOCAL COMPUTER DEALER, OR CALL TOLL FREE: 1-800-331-4411. IN OKLAHOMA CALL (918) 744-9844.

Visa or Mastercard Accepted Or Order By Mail

DEALERS WANTED

ATTENTION SOFTWARE PUBLISHERS — WRITE OR CALL FOR DETAILS ON SOFTWARE EVALUTION KIT.

KEYPORT 717® is a registered trademark of Polytel Computer Products Corp.
Apple, Apple II, II+ and IIIe are registered trademarks of Apple Computer Inc.
VisiCalc is a registered trademark of VisiCorp

POLYTEL

2121 S. Columbia
Tulsa, OK 74114

A.P.P.L.E. CONSULTANTS

John A. Hall, Jr.	Apple //e (East)	(617) 241-9050
Cecil Fretwell	Apple //e (MidWest)	(319) 236-0961
Bob Huelsdonk	Apple /// (West)	(206) 362-4910
Harry T. Hanson	Apple /// (East)	(201) 467-0712
Carmen Picardi	Apple /// (East)	(603) 895-4447
Stuart Yusem	Apple /// (East)	(609) 822-9306
Tim Smith	Apple /// (MidWest)	(389) 734-7451
Neil Clemmons	Agriculture	(314) 968-9618
Erv Edge	Assembly (Alaska)	(907) 337-7620
Dennis Baumgarte	Assembly (East)	(914) 298-8756
Murray Arnow	CPM (MidWest)	(312) 867-8000
Kevin Talbot	CPM (West)	(206) 232-7058
Steven E. Blair	Data Bases	(913) 782-5901
Jerry Rogan	Data Bases, Business	(319) 332-4166
William P. Steinberg	DOS (East)	(212) 921-0663
Art Schumer	DOS (North)	(701) 282-7907
Lee Meador	DOS (South)	(817) 469-6019
Wm. Reynolds III	DOS (West)	(503) 472-4028
Ted Perry	Education	(916) 485-1690
Cornelis Bongers	DOS, Applesoft (Europe)	010-525511 Ext. 3275
Alan Robinson	Fortran	(503) 757-8234
Jim Eatherly	Games	(202) 232-6046
Tedd Sperling	Geoscience	(517) 393-1296
Stan Axelrod	Hardware Design/I-O Control	(415) 563-1840
Tom Deal, M.D.	Handicapped Applications	(206) 634-2847
Randy Halvorson	Hardware Interfacing	(206) 783-9040
Ed So	Hi-Res	(408) 267-3448
Stowe Keller	Mathematics	(804) 979-4619
Fred Merchant, M.D.	Medical Group	(415) 921-7792
Stephen F. Lee	Music	(704) 289-4102
Tom Woteki	Pascal (East)	(202) 547-0984
Tim Smith	Pascal (MidWest)	(309) 734-7451
Paul E. Sleeper	Pascal (West)	(415) 494-1915
Bruce E. Sargent	Pascal (West)	(206) 488-1161
Dennis R. Cohen	Pascal (West)	(213) 956-8559
Earl Keyser	Pilot Language	(612) 429-5570
Randy Warrens	Pilot Language	(608) 837-9566
Jeff Sandys	Logo (Western U.S.)	(206) 784-3317
Hersch Pilloff	Stock Market	(301) 292-3100
Jim Hockenhull	Video Interfacing	(509) 334-4137
George L. Hefter	VisiCalc (East)	(617) 256-3082
Vince Aderente	VisiCalc (North)	(312) 289-9685
Dr. Bob Richard	VisiCalc (South)	(504) 466-0894
Brian Abrahams	VisiCalc (South Africa)	(Durban) 315442
Michael Ginsberg	Word Processing	(805) 735-8210

Call-A.P.P.L.E. consultants are unpaid volunteers and perform a valuable service. Before placing a call, please make sure you know the time zone you are calling to. No calls after 9pm, please.

We are always looking for more consultants. If you see an area that needs coverage, please drop us a note in the mail with your qualifications and availability.

ORDER PRODUCTS LISTED ON REVERSE

Membership # _____ Expiration date _____ Date _____

Name _____ Address _____

City _____ State/Country _____ Zip _____ Phone _____

PRODUCT		Unit Price	*FOREIGN SHIPPING WT.	AMOUNT
1				
2				
3				
4				
5				
6				

Mail to:
A.P.P.L.E.
21246 68th Ave. S.
Kent, WA 98032
(206) 872-2245

VISA Account No. _____ Exp. Date _____

MasterCard Account No. _____ Exp. Date _____

Subtotal _____

**Washington sales tax _____

TOTAL AMOUNT _____

CALL TOLL-FREE
1-800-426-3667 ORDERS ONLY
OUTSIDE WASHINGTON
ONLY (24 Hours)

***FOREIGN MEMBERS**
****WASH STATE RESIDENTS**
ADD 8.1% SALES TAX.

☐ APPLE // | //e PLEASE SPECIFY
☐ PLEASE USE PRODUCT NAME AND CODE WHEN ORDERING

FOREIGN MEMBERS: Please calculate postage based on weights in product descriptions.
(See postage table below for cost)

FOREIGN SHIPPING CHART

POSTAL TABLE		I West Indies Central American Columbia, Venezuela	II All South American Countries (except in Table I), Europe, N Africa	III Asia, Mid East, S. Africa Australia and Pacific Islands
0 oz	1 oz	$0.50	$0.60	$0.70
1	2	.76	.96	1.16
2	3	1.02	1.32	1.62
3	4	1.28	1.68	2.08
4	6	1.54	2.14	2.74
6	8	1.80	2.60	3.40
8	10	2.06	3.06	4.06
10	12	2.32	3.53	4.72
12	14	2.58	3.98	5.38
14	16	2.84	4.44	6.04
16	18	3.10	4.90	6.70
18	20	3.36	5.36	7.36
20	22	3.62	5.82	8.02
22	24	3.88	6.28	8.68
24	26	4.14	6.74	9.34
26	28	4.40	7.20	10.00
28	30	4.66	7.66	10.66
30	32	4.92	8.12	11.32
2 lbs	2.5	5.96	9.96	13.96
2.5	3.0	7.00	11.80	16.60
3.0	3.5	8.04	13.64	19.24
3.5	4.0	9.08	15.48	21.88
Each additional ½ pound over 4 pounds		add: 1.04	1.84	2.64

CANADA SHIPPING RATES

Up to:	
1	.20
2 oz	.37
3 oz	.54
4 oz	.71
5 oz	.88
6 oz	1.05
7 oz	1.22
8 oz	1.39
9 oz	1.56
10 oz	1.73
11 oz	1.90
12 oz	2.07
1 lb	2.58
2 lbs	3.10
3 lbs	4.10
4 lbs	5.10
5 lbs	6.10
6 lbs	7.10
7 lbs	8.10
8 lbs	9.10
9 lbs	10.10
10 lbs	11.10

For weights over 10 lbs, add $1.00 per pound.

Sorry, No Foreign Hardware Orders

A.P.P.L.E.™ PRODUCTS

AVAILABLE TO A.P.P.L.E MEMBERS ONLY

SOFTWARE

☐	SAMP	AMPERMANAGER	$22.50
☐	SAPF	AP FILE	$18.50
☐	SFLR	AP GRAPHICS "The Filler"	$17.00
☐	SAM 3	APMAIL II VERSION 3.0	$25.00
☐	SAPMPR	APMAIL.Pro	$35.00
☐	SAPMUP	AP MAIL UPDATE (Incl.Ser.#)	$10.00
☐	SAPT	AP TEST	$18.50
☐	SCAR	APPLESOFT CARPENTER//	$18.50
☐	SMLC	BIG MAC LC	$28.50
☐	SDISRE	LIFESAVER By Microlab	$21.00
☐	SDAN	DRIVE ANALYZER	$32.00
☐	SEAP	E Z PILOT	$28.50
☐	SDIC	DICE	$12.50
☐	STITWR	DIRECTORY TITLE WRITER	$15.00
☐	SD6B	UTILITY DISKPAK 6B-16	$18.50
☐	SHFI	HIGHER FONTS 1	$13.00
☐	SHTP	HIGHER TEXT PLUS	$27.50
☐	SIMAGI	IMAGINATOR	$60.00
☐	SMUL	MULTI-TRIEVE	$50.00
☐	SPGE	PROGRAM GLOBAL EDITOR	$18.50
☐	SPLE	PROGRAM LINE EDITOR	$15.00
☐	SPZAP	PROZAP	$25.00
☐	SS70	SOFT SEVENTY	$20.00
☐	SSYM	SYMBOL SYMON	$20.00
☐	STSS	THE SPREADSHEET	$75.00
☐	STSC	THE SPREADSHEET 1.0 to 2.0 data file converter	$5.00
☐	STUR	TURTLESOFT	$22.50

ValueSoft

☐	SCATKI	CATER KILLER	$12.50
☐	SDOSAN	DISK ANALYZER 1.3	$12.50
☐	SFARWO	FARWORD	$12.50
☐	SGRAFM	GRAFMAGIC	$12.50
☐	SGRABU	GRAPHICS BUILDER	$12.50
☐	SMICWT	MICROWRITER //e	$12.50
☐	SMORTG	MORTAGES, DEPREC. AND FINANCIAL CALCULATIONS	$12.50
☐	SSCRAM	SCRAMBLER	$12.50
☐	SGRADE	SCHOOLHOUSE GRADER	$12.50
☐	SZAPT	ZAPT	$12.50
☐	SWICWA	WICKERWARE #1	$12.50

APPLE /// PRODUCTS

☐	SWW3	WORD WEAVER ///	$75.00
☐	HDR3	APPLE /// DISK DRIVE	$285.00

HARDWARE

☐	HSRMON	AMBER SCREEN MONITOR By Leading Edge	$100.00
☐	HAPD	AP DRIVE	$235.00
☐	HAPC	AP CONTROLLER	$85.00
☐	HAKK	AP COOL KIT	$20.00
☐	HLC1	AP L CASE 1	$30.00
☐	HLC2	AP L CASE 2 DICE	$25.00
☐	HLC1A	w/ AP L CASE 1	$37.50
☐	HLC2B	w/ AP L CASE 2	$30.00
☐	H80C	APPLE //e 80 COLUMN CARD	$120.00
☐	HAUD	APPLESURANCE by Prometheus	$85.00
☐	HP16	AP RAM 16 K by Prometheus	$50.00
☐	HP28	AP RAM 128 K by Prometheus	$295.00
☐	HMPC	DUMPLING GX	$130.00
☐	HMPD	DUMPLING 64	$180.00
		GRAPHITTI by Prometheus	$85.00
		Please specify printer type:	
☐	HGRAEP	Epson, Panasonic	
☐	HGRACI	C. Itoh	
☐	HGRAOK	Okidata Prowriter, Microline, NEC	
☐	HGRAID	IDS (460, 560, Prism)	
☐	HGRAAP	Apple Dot Matrix	
☐	HDUST	APPLE DUST COVER	$8.00
☐	HP11	HOBBY PROTOTYPING BOARD	$21.00
☐	HHOC	HOME CONTROLLER by Prometheus	$40.00
☐	HPAD	KOALA PAD TOUCH TABLET	$95.00
☐	HJOYST	MACH II JOYSTICK By Hayes	$34.00
☐	HJMACH	MACH III JOYSTICK By Hayes	$44.00
☐	HJOMA3	MACH III JOYSTICK By Hayes (for //e)	$49.00
☐	HMOD	MODEM II By Multi-Tech	$195.00
☐	HPANMO	PANASONIC CT-160 COLOR MONITOR	$279.00
☐	HPANPR	PANASONIC KX1090 PRINTER	$325.00
☐	HSERIN	RS232 SERIAL INTERFACE CARD	$85.00
☐	HPANRI	PANASONIC P1090 RIBBONS	$14.00
☐	HPADDL	PADDLE-ADAPPLE By SCRG	$25.00
☐	HPRMOD	PROMODEM 1200	$450.00
☐	HPRT	PROWRITER PRINTER	$380.00
☐	HPRC	Prowriter Ribbon cartridges	$5.00
☐	HRIGNO	RIGHTNOTCH	$15.00
☐	HRPT	REPEATERRR	$23.00
☐	HSOFTC	SOFTCASE "FLASHCARD" By Synetix	$39.00
☐	HSSS	144 K SINGLE DRIVE	$275.00
☐	HSSU	144 K UPGRADE KIT	$200.00
☐	HSSD	288 K DUAL DRIVE	$500.00
☐	HFAN	SUPER FAN II Zener Ray	$79.00
☐	HTRC	TALLY MT 160L RIBBON	$12.50
☐	HTRI	TALLY MT 180L RIBBON	$17.00
☐	HULTRA	THE ULTRATERM By Videx	$310.00
☐	HTR120	TRANSTAR 120 Letter Quality Printer	$559.00
☐	HSUPMO	SUP R MOD By M&R	$60.00
☐	HURB	ULTRA ROM BOARD/EDITOR Firmware Enhancement System	$149.00
☐	HVRC1	VERSACARD by Prometheus with 6' par. cable	$150.00
☐	HVRC2M	with 6' male ser. cable	
☐	HVRC2F	with 6' female ser. cable	
☐	HVRC3	with 1' ser. cable	
☐	HVSLM	6' male ser. cable only	$20.00
☐	HVSLF	6' female ser. cable only	$20.00
☐	HVSS	1' ser. cable only	$20.00
		VERSACARD graphics PROMs	$30.00
☐	HVPREP	Epson, Panasonic	
☐	HVPRCI	C. Itoh	
☐	HVPRID	IDS	
☐	HVPROK	Okidata, ProWriter, NEC	
☐	HVPRAP	Apple D.M.	
☐	HVIE80	VIEWMASTER 80 COLUMN CARD By Applied Engineering	$149.00†

A.P.P.L.E. GAMES

☐	SKO	KOALA GRAMS SPELLING I	$31.50
☐	SSPI	SPIDER EATER	$24.00
☐	SCCO	COSMIC COMBAT	$12.50
☐	SD7	DISKPAK 7	$17.50
☐	SMCU	MUMMY'S CURSE	$12.50
☐	STHB	THUNDERBOMBS	$16.00
☐	SMIC	MICROBE	$30.00

A.P.P.L.E. LIBRARY

☐	PBAD	BENEATH APPLE DOS	$17.00
		Call-A.P.P.L.E. IN DEPTH	
☐	PID1	VOL I ALL ABOUT APPLESOFT	$7.00
☐	SIN1	VOL I Diskette	$16.50
☐	PID2	VOL II ALL ABOUT PASCAL	$11.00
☐	SIN2	VOL II Diskettes	$20.00
☐	PID3	VOL III ALL ABOUT DOS	$13.50
☐	SIN3	VOL III Diskettes	$15.00
		Call-A.P.P.L.E. BACK ISSUES	
☐	PPK4	BOUND SET (Call-A.P.P.L.E.) 1981	$18.50
☐	PPK5	BOUND SET (Call-A.P.P.L.E.) 1982	$18.50
		PEEKing at Call-A.P.P.L.E.	
☐	PPK2	VOL II 1979 Compendium	$10.00
☐	PPK3	VOL III 1980 Compendium	$15.00
		ANTHOLOGY DISKETTES (each)	$12.50
☐	SA1	VOL 1 (Jan thru Jun 1979)	
☐	SA2	VOL 2 (Jul thru Dec 1979)	
☐	SA3	VOL 3 (Jan thru Jun 1980)	
☐	SA4	VOL 4 (Jul thru Dec 1980)	
☐	SA5	VOL 5 (Jan Thru Jun 1981)	
☐	SA6	VOL 6 (Jul thru Dec 1981)	
☐	SA7	VOL 7 (Jan thru Jun 1982)	
☐	SA8	VOL 8 (Jul thru Dec 1982)	
☐	SA9A	VOL 9a (Jan to March 1983)	
☐	SA9B	VOL 9b (March to June 1983)	
☐	SA10A	VOL 10a (Jul to Sept 1983)	
☐	SA10B	VOL 10B (Oct to Dec 1983)	
☐	SPA1	Pascal '81	
☐	PACD	THE APPLE II CIRCUIT DESCRIPTION	$18.50
		THE POWER OF VISICALC	
☐	PVIS	VOL I	$8.50
☐	PVI2	VOL II	$8.50
☐	PU65	USING 6502 ASSEMBLY LANGUAGE	$17.00
☐	PASL	ASSEMBLY LINES: THE BOOK	$17.00
		WHAT'S WHERE IN THE APPLE	
☐	PWWA	COMPLETE VOLUME	$17.00
☐	PWW2	THE GUIDE	$8.00
☐	PKID	KIDS AND THE APPLE	$16.00
☐	SPDS	PUBLIC DOMAIN SOFTWARE (each disk volume)	$5.00
☐	SEAMON	EAMON ADVENTURES	$5.00
☐	PELA	THE ELEMENTARY APPLE	$11.00
☐	PWTA	WOZ TALKS (VHS video tape)	$27.50
☐	PIMAGS	IMAGINATOR SOURCE LISTING	$30.00

A.P.P.L.E. SERVICES AND ACCESSORIES

☐	HPAG	TYPING AID	$13.00
☐	FSOU	THE SOURCE SUBSCRIPTION	$60.00
☐	HFLS	FLIPSORT	$22.50
☐	HDBA	DISK BANKS	$4.50
		BLANK DISKETTES	
☐	HVEB	Verbatim Datalife (Box of 10)	$25.00
☐	HCDB	Control Data Corp (Box of 10)	$22.00
☐	HGEN	BULK BLANK PACK SS, DD, DISKS set of 100 with labels)	$150.00
☐	LID	LISA DISKETTES	$90.00
☐	HPOST	Call-A.P.P.L.E cover posters (specify month)	$9.95
☐	HSWSH	V-Neck Sweatshirts ☐ S ☐ M ☐ L ☐ XL	$13.95
☐	HTSH	T-Shirts ☐ S ☐ M ☐ L ☐ XL	$10.95
☐	FAPP	A.P.P.L.E. MEMBERSHIP ($25.00 first time application fee, $26.00 first years dues)	$51.00
☐	FREN26	A.P.P.L.E. MEMBERSHIP RENEWAL	$26.00

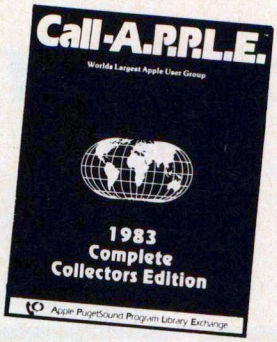
CALL-A.P.P.L.E. SPECIALS

☐	SEASPI	EAMON SPECIAL 1(Expires 4/30/84)	$22.50
☐	SEASP2	EAMON SPECIAL 2 (Expires 4/30/84)	$55.00
☐	SEASP3	EAMON SPECIAL 3 (Expires 4/30/84)	$175.00
☐	SPANTH	1983 ANTHOLOGY SPECIAL (With $50 order or new membership) (expires 5/31/84)	$35.00

* Indicates Introductory Price
† Subject to Availability

Use Order Form on Reverse

HOT OFF THE PRESS!



The Call—A.P.P.L.E. 1983 Complete Collectors Edition

12 issues from 1983
three-ring punched
with a sturdy outer cover
and index.

Member price $18.50
Product code PPK6
Foreign shipping wt: 7 lbs.

Four Single "Anthology" Diskettes

Containing all 1983
Call—A.P.P.L.E. Programs

Member price $12.50 ea.

Product codes: Jan thru Mar SA9A,
Apr thru June SA9B, July thru Sept SA10A,
Oct thru Dec SA10B
Foreign shipping weight: 2.0~~2~~ each

TOTAL COST
$68.50

BUT . . . if you join now, OR order $50 worth of A.P.P.L.E. products, you can get ALL of the above for just:

$35.00
member price

1983 Anthology Special

Product code SPANTH
Foreign shipping weight: 8 lbs.

Offer Expires 5/31/84
Wash. State residents add 8.1% sales tax.
24 Hour toll free (800) 426-3667 (outside Washington)

A.P.P.L.E. ORDERS
21246 - 68th Ave. So.
Kent, WA 98032

A Graphics Tool Kit

(PART 1)

randi rost

HELLO everyone, and welcome to the first of what I hope will become a regular series in the pages of *Call -A.P.P.L.E.* — the Graphics Toolkit. If all of you dutifully type in the program listings every month, you'll soon have a useful library of machine language graphics routines that you can draw upon for all sorts of applications. Bear with me as I get some philosophical introductory remarks out of the way.

I. Introduction

Stating the problem to be solved before actually going ahead and solving the problem always helps to focus one's thinking. This is referred to, logically enough, as the problem statement. What is the goal of this series? The main goal is to share with *Call -A.P.P.L.E.* readers some machine language routines that will be useful for graphics applications such as animation and games. Along the way, I hope to explain a few of the intricacies of graphics on the Apple as well as discuss some general graphics algorithms. During the course of this series, we'll be developing routines to draw lines, plot block shapes, and all sorts of wonderful things. I'll also take time out every so often to encourage some progressive thinking on software engineering. Those who take my words to heart will only be ahead of the game when it comes to producing 1980s-style software.

What will you need in order to get something out of this series? An assembler is necessary for those of you who wish to use the routines that will be presented. I recommend the Big Mac assembler from A.P.P.L.E., not because it's from A.P.P.L.E., but because it's the best. It also happens to be a steal if you're an A.P.P.L.E. member. I also hope that those who don't want to type in the listings, pick up a few good ideas just by reading the text of the articles.

I will be presuming some knowledge of 6502 machine language and of graphics on the Apple in general. The routines presented will either be assembly language or Applesoft. If you have any qualms about any of this, don't worry just yet. Everyone should go back and review Loy Spurlock's excellent article "Understanding High-Res Graphics" in the Jan. 1980 issue of *Call -A.P.P.L.E.* or the 1980 anthology volume. Many other recent articles can provide some general information on Apple graphics. These include Mark Pelczarski's "Graphically Speaking" series or Roger Wagner's "Assembly Lines" in past issues of *Softalk,* David Lubar's series "The Graph Paper" in *Creative Computing* and Ken Williams treatment of the topic in *Softline.*

I would also encourage questions and comments to be directed to The Graphics Toolkit, C/O *Call -A.P.P.L.E.* The inescapable lead times needed in producing a magazine will make answering questions in future columns somewhat difficult, but it still seems like the most reasonable approach. Any ideas on topics for future columns will also be taken into consideration. This magazine is here to provide you with what you want, so please contribute feedback.

II. Tools and Toolkits

What is a tool? A tool is something that is used by a craftsperson or a laborer to make their work easier. To call something a tool also often implies reusability, that is, the tool can be used every time the circumstances warrant its use. A toolkit in the software sense then might simply be a collection of subroutines that can simplify a specific programming task. In this case, we have chosen graphics as our applications area. Reusability implies that this collection of routines will be well documented and easy enough to use so that it can be picked up a year later and still be useable.

There were five major goals established in the development of the routines that will appear in this series. I'll list them in the order in which they influenced the design of the overall package and leave it up to you to see how well the goals were met. The goals were:

1) Flexibility/Generality
2) Speed
3) Ease of use
4) Modularity
5) Compactness

Surprised to see speed ranked number two? Actually the first three were all very close in terms of importance, and the last two were somewhat more secondary considerations. These routines all stemmed out of a desire to develop the tools necessary to write an arcade-quality game on the Apple. At the same time, there was a strong desire to make the routines as easy to use as possible so that friends and relatives with Apples could use them as well. Feel free to use these routines in any of your own programs, but please contact me through A.P.P.L.E. before using any of the routines in commercial programs, as the routines are copyrighted by A.P.P.L.E.

III. Apple Graphics

Ok, let's get to the heart of the matter. You will recall that the primary High-Res page lives in the memory range $2000-$3FFF (all numbers preceded by dollar signs are

in hexadecimal) and High-Res page two resides in range $4000-$5FFF. Each row of 280 pixels on the High-Res screen corresponds to a 40 byte sequence in memory. The address of the first byte in the sequence depends on the y-value, or row number, and whether the row is part of High-Res page one or High-Res page two.

Unfortunately, the rows of the High-Res screen are not mapped sequentially into the High-Res memory area, but in a somewhat complicated non-sequential pattern. This pattern is visible whenever you type the Applesoft command "HGR" to clear a cluttered graphics screen. The screen is not cleared in order from top to bottom but appears to be divided into bands of about eight lines each that are erased simultaneously. In fact, the value for black *is* being loaded sequentially into every location from $2000-$3FFF and the effect you are seeing demonstrates that any given row on the screen does *not* immediately precede in the memory scheme the row directly beneath it on the screen. Spurlock derived a formula for calculating the address of the first byte of row given only its row number (y-value) in the *Call -A.P.P.L.E* article mentioned earlier. More on this topic a little later.

Once you have found the address of the first byte in the row, the rest is a little easier. Since there are 280 pixels in a row and 40 bytes in which to store the on/off information for one row, it seems reasonable that each byte will contain on/off information for seven pixels. This is exactly what happens. The eighth bit in each byte, the most significant bit, is used as a color flag for the seven pixels in the byte.

The pattern contained in a specific byte determines the color and on/off information for seven pixels on the screen. If all seven pixel bits are zeros, (off) then we have black in those seven pixels on the screen. Since the color bit may be either one or zero, we have two ways of representing black. The same is true for white. All seven pixel bits can be ones (on) to display a short white line on the screen, and the color bit can be either on or off, giving us two ways to create white.

Colors are formed in a similar manner. It turns out that alternating pixels in an on-off-on-off pattern will cause a color other than white or black to be created. Since the pattern can be either on-off-on-etc. or off-on-off-etc. and the color bit can be either on or off, we have four more colors (violet, green, orange, and blue) that may be used. Since it takes an on-off pair to make a dot on the screen in one of these colors, the resolution of the High-Res screen is really only 140x192 for colors other than white and black. This also means that vertical lines in these colors will cause problems. Green and orange vertical lines can be seen in odd columns but not in even columns, and for blue and violet the opposite is true.

. . . the resolution of the high-res screen is really only 140 x 192.

Allowing the color flag to control the color of seven pixels causes another well-known problem: "unexpected" color changes when drawing lines or shapes using a color whose the color flag is different than the color flag of the background. The new pixel is plotted as one of the seven on/off bits in the byte, and the color flag is changed. The result is that the other six bits in the byte now have a different color flag than before and may be displayed as a completely different color. There really is no solution to this problem, other than precluding the problem a priori by choosing background and foreground colors that do not interfere in such a manner.

IV. Lookups

What is the fastest way to multiply two numbers together on a computer? Before you start describing efficient multiplication algorithms, think a little harder. On nearly any computer, the fastest way to multiply two numbers is to use a large table to store all the answers, and use the two numbers as indices into the table. Once the proper location in the table is found, it's simply a matter of reading the answer. This method is commonly referred to as "table lookup." The sheer size of this table for multiplication of two arbitrary numbers makes it infeasible in this case, but it can be used to great advantage in speeding up calculations if the table does not have to be very large.

The address of the first byte in any particular row of the High-Res screen can be calculated given only the row number, but it turns out that a 20-30 line machine language routine is needed. Since there are only 192 rows on the High-Res screen, it appears we can save a good deal of calculation with a table that's not too large. In this case we need two 192-byte tables, one for the high byte of the address and one for the low byte. Doing two indexed memory accesses will give us the address of any row. This will speed up the row address calculation by more than a factor of 20!

Listing 1 contains an Applesoft program that will POKE into memory all the tables that will be used by all the graphics routines to be presented in this series. The first table contains the high byte base address for each of the 192 rows. Adding $20 to this value will result in an address on High-Res page one, and adding $40 yields an address on page two. The second table contains the low byte of the base address for the row.

Following these two tables are two very short tables. The first is called the color mask table and contains eight bytes, one for each of the eight possible High-Res colors. Each byte in this table contains the bit pattern (on-off-on. . . or whatever) that is needed to render that particular color on the display. Following that is a seven-byte table called the bit-position table. Each byte in this table contains a one in the color flag bit and a one in the bit which corresponds to its location within the table. For instance the first entry has bit zero set, the second entry has bit one set, etc.

The fifth and sixth tables also speed up computations. They contain values for locating the specific byte within a row and the bit within that byte for a specific x-coordinate on the screen. The fifth table contains values of x/7 since dividing by seven is a time-consuming operation compared to a table lookup. The sixth table contains values for (x mod 7) + 1, and we'll see how those values will be useful later on in the series.

```
]LIST

100  REM  -- INITIALIZATION --
110 A = 8192
200  REM  -- POKE Y-HI VALUES --
210  FOR I = 1 TO 3
220 :: FOR J = 0 TO 3
230 :::: FOR K = 1 TO 2
240 :::::: FOR L = J TO J + 28 STEP
    4
250 :::::::: POKE A,L
260 ::::::::A = A + 1
270  NEXT L,K,J,I
300  REM  -- POKE Y-LO VALUES --
310  FOR I = 1 TO 3
320 :: READ YLO(1),YLO(2)
330 :: FOR J = 1 TO 4
340 :::: FOR K = 1 TO 2
350 :::::: FOR L = 1 TO 8
360 :::::::: POKE A,YLO(K)
370 ::::::::A = A + 1
380  NEXT L,K,J,I
390  DATA  0,128,40,168,80,208
400  REM  -- POKE COLOR MASK TABL
    E --
410  FOR I = 1 TO 8
420 :: READ COLMASK
430 :: POKE A,COLMASK
440 ::A = A + 1
450  NEXT I
460  DATA  0,42,85,127,128,170,21
    3,255
500  REM  -- POKE BIT POSITION TA
    BLE --
510  FOR I = 0 TO 6
520 :: POKE A,128 + 2 ^ I
530 ::A = A + 1
540  NEXT I
600  REM  -- POKE X/7 TABLE --
610  FOR I = 0 TO 57
620 :: FOR J = 1 TO 7
630 :::: POKE A,I
640 ::::A = A + 1
650  NEXT J,I
700  REM  -- POKE (X MOD 7) + 1 T
    ABLE --
710  FOR I = 0 TO 57
720 :: FOR J = 1 TO 7
730 :::: POKE A,J
740 ::::A = A + 1
750  NEXT J,I
800  REM  -- CHECK FOR ACCURACY -
    -
810 SUM = 0
820  FOR I = 8192 TO A - 1
830 ::SUM = SUM +  PEEK (I)
840 :: IF SUM > 255 THEN SUM = SU
    M - 255
850  NEXT I
855  TEXT : HOME : VTAB 8
860  PRINT "CHECKSUM IS -- ";SUM
870  PRINT
880  IF SUM <  > 187 THEN  PRINT
     "ERROR -- CHECKSUM SHOULD EQ
     UAL 187.": PRINT "PLEASE REC
     HECK YOUR PROGRAM.": GOTO 10
     00
900  PRINT  CHR$ (4);"BSAVE YTABL
     E,A$2000,L$460"
1000  END
```

Listing 1

The program in listing 1 will try to do a checksum on the POKEd values, and if the result is successful, it will save the tables onto your disk under the name YTABLE. As you can see, this file contains only data values and so is completely relocatable. It can be loaded into memory anywhere there is enough space, and is just over 1K bytes long.

V. Initialization

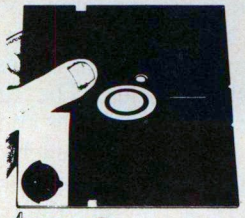
In just about any programming task, there are a few activities that must be carried out under the heading of initialization, and this is no exception. We have created a set of lookup tables that are completely relocatable, but our other routines will need to know where those tables have been loaded. The initialization routine in Listing 2 will take care of installing the necessary pointers to our lookup tables.

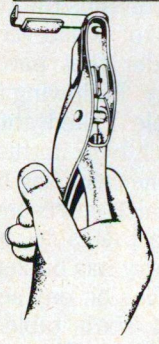
The initialization routine should be called with the address of the YTABLE in the A and X registers. Right from the start let's adopt the convention that whenever it's necessary to pass an address to a routine, it will be passed with the high byte of the address in the A register and the low byte in X. This will keep address-passing consistent and easy to remember for all of our routines. In this case, if our YTABLE was loaded at $1000, the first thing we'd need in our program is a JSR INIT after loading the A register with a $10 and the X register with zero. The INIT routine will then set up the necessary pointers so that later routines will know where to find the lookup tables. Later on in the series we'll be adding a couple more things to the initialization routine.

VI. A Tool or Three

Now that we have the groundwork laid out let's get to work. The HGR and HGR2 commands perform very similar functions in Applesoft. HGR displays and clears High-Res page one and sets the current plotting color to black ($00). HGR2 does the same thing but on High-Res page two. The only other difference between the two is that HGR sets mixed text and graphics mode, and HGR2 sets full-screen graphics mode. Another potentially useful function missing from Applesoft is one that would clear the current High-Res page to any desired color.

***REDUCES DISKETTE COST 50%!**
DOUBLES DISKETTE STORAGE SPACE





Owners of 5¼" *single *"read/write"* head disk drives can immediately double diskette storage space by using ***NIBBLE NOTCH I*** or ***II.*** The back of a single sided diskette is burnished. To use it you need a *"write enable notch,"* and some also need in addition, an *"index hole."* ***NIBBLE NOTCH I*** and ***II*** are precision engineered tools designed exclusively for this purpose.

NIBBLE NOTCH I

(Cuts square Write Enable Notch) For users of Apple, Franklin, Atari, Commodore and most other soft sectored systems.

only **$14.95** each
add $1.50 each P&H / ($4.50 foreign P&H)

NIBBLE NOTCH II

(Cuts square Write Enable Notch and ¼ inch round *"index hole"*) for owners of IBM, TRS 80 I & III, Osborne, Kaypro and others needing *"index hole"*, and all other *"hard sectored"* systems.

only **$21.90** each
add $2.50 each P&H / ($6.50 each foreign P&H)
Florida Residents Add 5% Sales Tax

• IT'S A MONEY SAVER! •

Dealer inquiries are invited.

ORDER TODAY!
SEND CHECK OR MONEY ORDER TO:

NIBBLE NOTCH™
DIVISION OF CORTRAN INTERNATIONAL
4211 N.W. 75 TERRACE • DEPT. 76 • LAUDERHILL, FL 33319
PAT. PEND. ALL TRADEMARKS ARE ACKNOWLEDGED

Circle 4 on Reader Service Card

```
A S M

                        ********************************
                        *                              *
                        *  HI-RES GRAPHICS SUBROUTINES *
                        *                              *
                        *      by  Randi J. Rost       *
                        *                              *
                        * Call -A.P.P.L.E. : Apr. 1984 *
                        *      All rights reserved     *
                        *                              *
                        ********************************
                        *
                        *
                        *
                        * The INIT subroutine should be called with
                        *  the lo byte of the YTABLE address in X and
                        *  the hi byte in A.  Other ptrs are then installed.
                        *
                        *
                                ORG   $803
                        YTABL   =     $EC
                        YTABH   =     $EE
                        POSTABL =     $4A
                        XDIVTBL =     $4C
                        XMODTBL =     $D0
                        COLTABL =     $FE
0803: 85 EF             INIT    STA   YTABH+1
0805: 86 EE                     STX   YTABH
0807: 8A                        TXA
0808: 18                        CLC
0809: 69 C0                     ADC   #$C0        ;Get starting loc for
080B: 85 EC                     STA   YTABL       ;Y-lo values table
080D: A9 00                     LDA   #$00
                        *
080F: 65 EF                     ADC   YTABH+1
0811: 85 ED                     STA   YTABL+1
0813: 18                        CLC
0814: A5 EC                     LDA   YTABL
0816: 69 C0                     ADC   #$C0        ;Get starting loc for
0818: 85 FE                     STA   COLTABL     ;color masking table
081A: A9 00                     LDA   #$00
081C: 65 ED                     ADC   YTABL+1
081E: 85 FF                     STA   COLTABL+1
0820: 18                        CLC
0821: A5 FE                     LDA   COLTABL
0823: 69 08                     ADC   #$08        ;Get starting loc for
0825: 85 4A                     STA   POSTABL     ;bit position table
0827: A9 00                     LDA   #$00
0829: 65 FF                     ADC   COLTABL+1
082B: 85 4B                     STA   POSTABL+1
082D: 18                        CLC
082E: A5 4A                     LDA   POSTABL
0830: 69 07                     ADC   #$07        ;Get starting loc for
0832: 85 4C                     STA   XDIVTBL     ;X/7 table
0834: A9 00                     LDA   #$00
0836: 65 4B                     ADC   POSTABL+1
0838: 85 4D                     STA   XDIVTBL+1
083A: 18                        CLC
083B: A5 4C                     LDA   XDIVTBL
083D: 69 96                     ADC   #$96        ;Get starting loc for
083F: 85 D0                     STA   XMODTBL     ;(X MOD 14)/2 table
0841: A9 01                     LDA   #$01
0843: 65 4D                     ADC   XDIVTBL+1
0845: 85 D1                     STA   XMODTBL+1
0847: 60                        RTS

--End assembly, 69 bytes, Errors: 0
```

Listing 2

Is it possible that we could kill all three birds with one stone?

Listing 3 shows that the answer is yes. By carefully coding the routine and providing three separate entry points, we can accomplish all three tasks with the same routine. Again for consistency, we'll choose to make both the HGR and HGR2 routines set full-screen graphics mode. Let's take a close look at the routines contained in Listing 3. The instructions at the HGR2 entry point cause page two of High-Res graphics to be displayed and the A register to be loaded with a value of $40 before branching around the code for the HGR entry point. For HGR, a value of $20 is loaded into A and High-Res page one is displayed.

We have now reached the label HCLEAR with either a $20 or a $40 in the A register. Recall that High-Res page one starts at $2000 and page two starts at $4000. The value in A then is simply the high byte of the start address of the graphics page to be manipulated. This value is stored immediately in the location HGRPG. HGRPG is a very special page zero location that will always point to the High-Res page currently being read or written (but not necessarily displayed). It will be used extensively by other routines. Next we tweak all the switches necessary to set up full page, High-Res graphics, and then load A with the color to be used in filling the screen – black0.

Now we're at the third of our function entry points. If we jump directly to BKGND with the color (0–7) to be used in filling the screen in the A register, we can skip all the set-up work performed by HGR and HGR2 and take advantage of the same code they use to fill the screen with zeros (black0). At this point we transfer the contents of A into Y and do an indirect indexed load of the A register using COLTABL. COLTABL is another page zero location, set up by the INIT routine to point to the start of the color mask table. This loads the appropriate color mask byte for the specified color from our lookup table. The color mask value becomes our current plotting color and is stored in two locations, COLOR and COL1 where it can be found later. It's also stored in the location SHFTCHK where it can be found and used by the SHFTCOL routine.

Remember when we were discussing the on-off-on bit patterns earlier? We discovered that there are seven pixels per byte, right? This implies that we need to be careful in order to maintain a pattern of on-off-on through several bytes. If we have 1010101 for the first byte *and* for the second byte, the concatenated result is 10101011010101, clearly not what we want. What we need is a 1010101 followed by a 0101010 and so on. This one-bit shift is necessary to maintain the proper color masking pattern for all the bytes in a row and is precisely the function performed by the SHFTCOL routine.

Upon returning from SHFTCOL, the shifted color mask is loaded from SHFTCHK and stored in COL2. We now have everything we need and the rest of the routine is straightforward. A big loop is set up and all we need to do is copy COL1 into all the even bytes and COL2 into all the odd bytes of the High-Res screen indicated by HGRPG.

VII. A Demo Program

Let's write a little demo program and finish up for this month. The first thing we need to do is decide where to put our lookup tables, so for lack of a better location let's assume that they will be loaded at $1B00 (just below High-Res page one). For this reason, you will need to BLOAD YTABLE, A$1B00 as well as loading the INIT and HGR routines at their

```
A S M
                        1     *
                        2     * The HGR and HGR2 subroutines function almost
                        3     *  identically to their Applesoft counterparts.
                        4     * No arguments need be passed to HGR and HGR2.
                        5     * To use BKGND, store the color (0-7) to clear the
                        6     *  screen to in A, then JSR BKGND.
                        7     *
                        8               ORG   $85B
                        9
                        10    SHFTCHK   =     $1C
                        11    TEMPPTR   =     $26
                        12    COLOR     =     $E4
                        13    HGRPG     =     $E6
                        14    COLTABL   =     $FE
                        15    DISPGR    =     $C050
                        16    NOMIX     =     $C052
                        17    DISPG1    =     $C054
                        18    DISPG2    =     $C055
                        19    HIRES     =     $C057
                        20
                        21    COL1      DS    1
                        22    COL2      DS    1
                        23    TEMPCNT   DS    1
                        24
085E: 2C 55 C0          25    HGR2      BIT   DISPG2       ;Display hi-res pg 2
0861: A9 40             26              LDA   #$40         ;Hi byte of base adrs for pg2
0863: D0 05             27              BNE   HCLEAR
0865: A9 20             28    HGR       LDA   #$20         ;Hibyt of base adrs for pg1
0867: 2C 54 C0          29              BIT   DISPG1       ;Display hi-res pg1
086A: 2C 52 C0          30    HCLEAR    BIT   NOMIX        ;Set all graphics mode
086D: 85 E6             31              STA   HGRPG        ;Store base adrs in HGRpg
086F: AD 57 C0          32              LDA   HIRES        ;Set HIRES mode
0872: AD 50 C0          33              LDA   DISPGR       ;Display graphics page
0875: A9 00             34              LDA   #$00         ;COLOR=Black0
0877: A8                35    BKGND     TAY
0878: B1 FE             36              LDA   (COLTABL),Y
087A: 85 E4             37              STA   COLOR
087C: 8D 5B 08          38              STA   COL1
087F: 85 1C             39              STA   SHFTCHK
0881: 20 B0 08          40              JSR   SHFTCOL      ;Shift color masking byte
0884: A5 1C             41              LDA   SHFTCHK
0886: 8D 5C 08          42              STA   COL2
0889: A5 E6             43              LDA   HGRPG        ;Clear correct HGR page
088B: 85 27             44              STA   TEMPPTR+1
088D: A9 00             45              LDA   #0
088F: 85 26             46              STA   TEMPPTR
0891: A9 20             47              LDA   #$20         ;Hi byte of loop limit
0893: 8D 5D 08          48              STA   TEMPCNT
0896: A0 FF             49    FILLSCR   LDY   #$FF
0898: AD 5B 08          50    LOOP1A    LDA   COL1         ;Store col1 in even bytes
089B: 91 26             51              STA   (TEMPPTR),Y
089D: 88                52              DEY
089E: AD 5C 08          53              LDA   COL2         ;Store col2 in odd bytes
08A1: 91 26             54              STA   (TEMPPTR),Y
08A3: 88                55              DEY
08A4: C0 FF             56              CPY   #$FF
08A6: D0 F0             57              BNE   LOOP1A       ;Loop back for more
08A8: E6 27             58              INC   TEMPPTR+1    ;Increment hi byte of ptr
08AA: CE 5D 08          59              DEC   TEMPCNT      ;Dec. hi byte of loop count
08AD: D0 E7             60              BNE   FILLSCR
08AF: 60                61              RTS
08B0: 0A                62    SHFTCOL   ASL
08B1: C9 C0             63              CMP   #$C0         ;Shift color masking
08B3: 10 06             64              BPL   DONE         ;byte routine for
08B5: A5 1C             65              LDA   SHFTCHK      ;odd and even
08B7: 49 7F             66              EOR   #$7F         ;screen bytes
08B9: 85 1C             67              STA   SHFTCHK
08BB: 60                68    DONE      RTS

--End assembly, 97 bytes, Errors: 0
```

Listing 3

ORG addresses prior to BRUNning this demo. The first thing we need to do in the demo program is call INIT with the address of the YTABLE in A and X. This is taken care of by the first three lines of the program in Listing 4.

During the course of this series, we'll be developing more tools that can be used to solve graphics applications problems on the Apple. Each month we'll present a new routine or two. You can decide for yourself how to organize this library of useful subroutines. If you type in all of them, the ORG addresses will all fit nicely together and you'll have a good-sized library, that, with the lookup tables, just fits under High-Res page one. This leaves the largest contiguous memory block in the Apple, from $6000-$9600, available for applications programs. Alternatively, you can use only those routines you need and assemble them anywhere in memory, but be careful about declaring all the addresses and variables needed to make each routine work. Some, like INIT, are needed for any of the routines to work, and nearly all will depend on the lookup tables. Take some time to absorb all this, and we'll dive into it again next month!

```
: A S M
                        1    *
                        2    * Title: Demo Program 1
                        3    * Function: Test INIT, HGR, HGR2 and BKGND routines
                        4    *
                        5            ORG   $6000
                        6
                        7    INIT    EQU   $803
                        8    HGR2    EQU   $85E
                        9    HGR     EQU   $865
                       10    BKGND   EQU   $877
                       11
6000: A9 1B             12    DEMO1   LDA   #$1B       ;Load A with YTABLE adr hi
6002: A2 00             13            LDX   #$00       ;Load X with YTABLE adr lo
6004: 20 03 08          14            JSR   INIT       ;Set up lookup table ptrs
6007: 20 65 08          15            JSR   HGR        ;Init hi-res graphics on page 1
600A: A9 00             16            LDA   #$00       ;Starting color is 0 (black0)
600C: 8D 1F 60          17            STA   COLORNO    ;Save it for later use
600F: AD 1F 60          18    LOOP    LDA   COLORNO    ;Load color number for BKGND
6012: C9 08             19            CMP   #$08       ;See if we're done
6014: F0 08             20            BEQ   DONE       ;If so, branch to DONE
6016: 20 77 08          21            JSR   BKGND      ;Else do BKGND one more time
6019: EE 1F 60          22            INC   COLORNO    ;Do next color
601C: D0 F1             23            BNE   LOOP       ;Branch always to LOOP
601E: 60                24    DONE    RTS
                        25
                        26    COLORNO DS    1

--End assembly, 32 bytes, Errors: 0
```

Listing 4

At this point, we can invoke any of the routines in our toolkit. Since we need to clear the screen and initialize High-Res graphics mode, the first thing we do is a JSR HGR. Next we'll set up a short loop and do a JSR BKGND for each of the eight High-Res colors. This will show how rapidly the High-Res screen can be cleared to any color you like. The implementation of the lookup tables makes our routine significantly faster than the standard Applesoft HGR routine. You may also want to change the JSR HGR to a JSR HGR2 to verify that the BKGND routine will work on High-Res page 2 as well.

VII. What To Expect Dept.

Where do we go from here? The difference between the routines in listings 2 and 3 and the routine in listing 4 is that the first two are *tools* and the last is an *application* that utilizes those tools. For the demo program, we identified a problem to solve, namely, to sequentially clear the graphics screen to each of the eight High-Res colors. The tools we developed earlier helped us to solve our problem.

"TELEDELIVERY"?

- SOFTWARE ON DEMAND
- FASTEST DELIVERY ANYWHERE
- 24-HOUR ACCESS, WORLDWIDE
- SAVE TIME, TRAVEL, MONEY
- EXCELLENT CUSTOMER SUPPORT

TRY IT NOW!

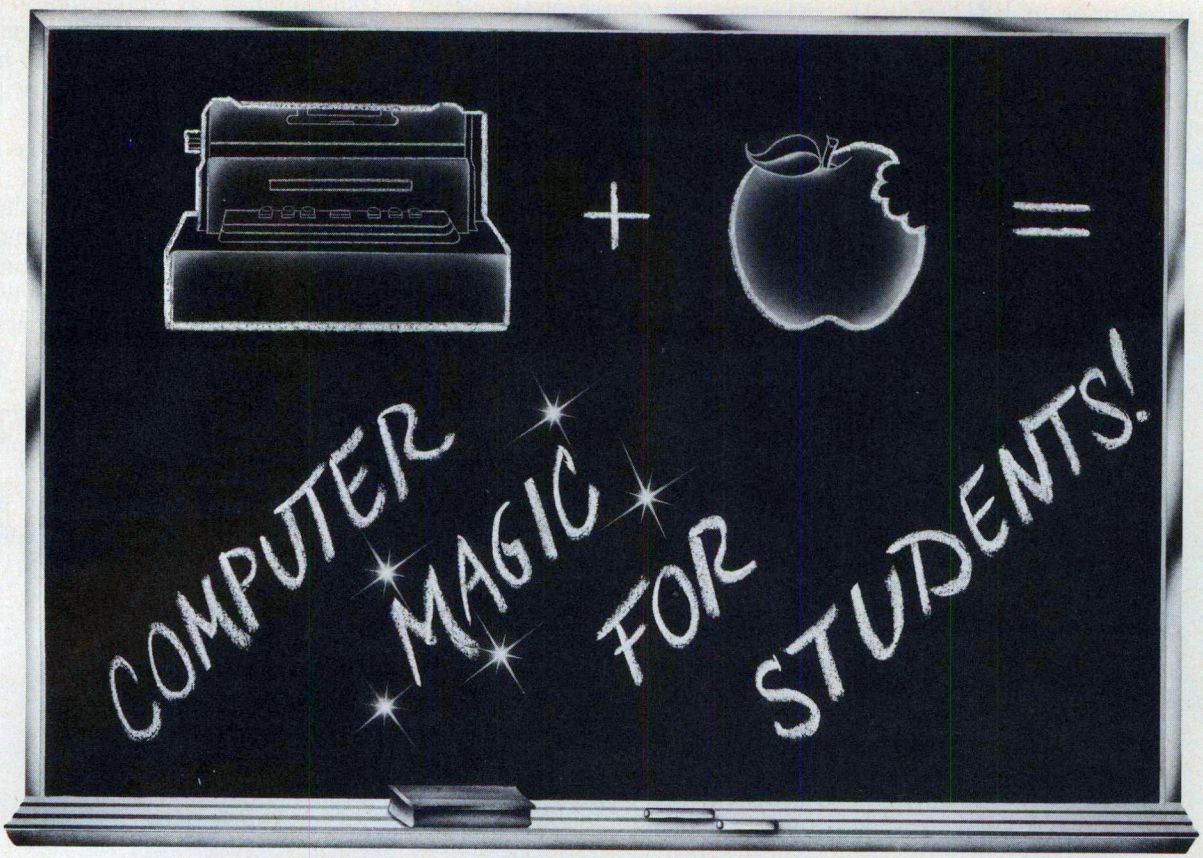
CALL, VIA MODEM: (213) 516-9432

AND EXPERIENCE THE FUTURE.

TELEPHONE SOFTWARE CONNECTION INC

POST OFFICE BOX 6548 - TORRANCE, CALIFORNIA 90504

Circle 6 on Reader Service Card

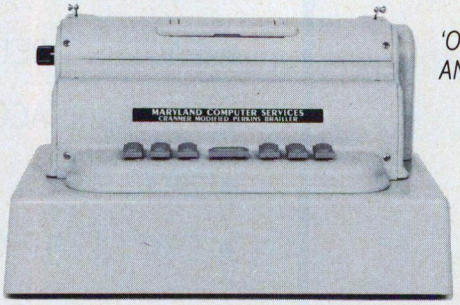


How does a professor who can't read or write braille communicate on paper with a student who is blind? Easy! Take one Apple Computer or other micro, hook it up to the new computerized Perkins Brailler manufactured by Maryland Computer Services and the results "Speak for themselves".

MCS' low cost braille printer, "Perky", offers students an easy and affordable way to access information from the computer.

Similar to a printer, Perky will plug into the Apple so anything can be printed out in braille . . . class notes, homework assignments and term papers. And that's without any special skills required or the traditional delays spent using a conventional brailler! Any of this information is transferrable from Perky to a standard tape for permanent storage. Students can keep a library of tapes and recall any information to be printed for reference. And because of Perky's editing capabilities, typos are quickly corrected without retyping the entire paper!

Teachers find Perky an asset in and out of the classroom. As a time saving device it allows teachers to communicate with students regardless of conflicting schedules. Tests and instructional materials can be typed into the Apple anytime for braille translation at the student's convenience.



*'Perky'—Cranmer Modified Perkins Brailler

Another Plus—Perky is the only full graphics printer. This unique feature provides the capability to produce maps, math symbols, and any figure, in tactile form.

So students, don't take an Apple to your teacher, take a Perky to their Apple. Another A+ idea from MCS.

'OUR PRODUCTS SPEAK FOR THEMSELVES AND WE STAND BEHIND WHAT THEY SAY'

MARYLAND COMPUTER SERVICES, INC.

2010 Rock Spring Road
Forest Hill, MD 21050
(301) 879-3366 • Telex: 701440

MARYLAND COMPUTER SERVICES, INC.

For further information please send to:
Maryland Computer Services, Inc.
2010 Rock Spring Road
Forest Hill, MD 21050

NAME

TITLE

ORGANIZATION/COMPANY

ADDRESS

CITY STATE ZIP

PHONE

GSA Schedule—Contract Number GS00K8401S583.

Adding XPLOT to Applesoft

mark harris

ONE deficiency of Applesoft graphics which has caused me considerable inconvenience is the lack of a line-drawing mode which doesn't destroy the background. Several other languages for the Apple (e.g. Pascal and GraFORTH) have a "reverse" mode in which the state of each pixel crossed by a line is inverted, so that reversing the same figure twice restores the original background. Applesoft does let the user reverse shapes with XDRAW, but shape tables are of little value in complex line drawings. What is needed is an "XPLOT" instruction which will duplicate the functions of HPLOT in a reverse mode. The assembly listing below provides such a function.

The key to a reverse mode is the 6502 instruction EOR, which takes the "exclusive-or" of the accumulator with the memory specified. This operation preserves information in the sense that if the accumulator is exclusive-or'ed with the same memory twice, the original value in the accumulator is restored. The ORA and AND operations do not have this property: for example, once the AND of the accumulator and memory has been performed, it is impossible to determine the previous contents of the accumulator. The line-drawing routines in Applesoft essentially OR the user's line with screen memory; to XPLOT we simply EOR instead. It's a pity that this feature was not included; it would have taken no more than a few extra bytes. But since we can't change the existing ROM, we must re-write a larger chunk of the code.

Many ROM routines are so intertwined with all of Applesoft that it is not possible to extract a small portion for modification. A disassembly of the HPLOT routine (with Sourceror from the Big Mac diskette) encourag-

```
: A S M
                    ********************************
                    *                              *
                    *   XPLOT   by Mark Harris     *
                    *                              *
                    * Call -A.P.P.L.E. : Apr. 1984 *
                    *                              *
                    * Example of use:              *
                    * XP = 768                     *
                    * CALL XP.100.100              *
                    *  Xplots single point.        *
                    * CALL XP.100.100 TO 200.100   *
                    *  Xplots line.                *
                    * CALL XP. TO 100.100          *
                    *  Xplots line from last       *
                    *  point plotted.              *
                    *                              *
                    ********************************

                                ORG   $300
                    *
                    LFTRT       =     $F465
                    UPDOWN      =     $F4D3
                    HPOSN       =     $F411
                    CHKCOM      =     $DEBE
                    TO          =     $C1
                    HFNS        =     $F6B9
                    CHRGOT      =     $B7
                    SYNCHR      =     $DEC0
                    DSCTMP      =     $9D
                    CHRGET      =     $B1
                    *

0300: 20 BE DE      XPLOT       JSR   CHKCOM
0303: C9 C1                     CMP   #TO
0305: F0 0D                     BEQ   XP3
0307: 20 B9 F6                  JSR   HFNS
030A: 20 A7 03                  JSR   HPLOT0
030D: 20 B7 00      XP2         JSR   CHRGOT
0310: C9 C1                     CMP   #TO
0312: D0 1B                     BNE   RET
0314: A6 E0         XP3         LDX   $E0
0316: A4 E1                     LDY   $E1
0318: A5 E2                     LDA   $E2
031A: 20 A7 03                  JSR   HPLOT0
031D: 20 B1 00                  JSR   CHRGET
0320: 20 B9 F6                  JSR   HFNS
0323: 84 9D                     STY   DSCTMP
0325: A8                        TAY
0326: 8A                        TXA
0327: A6 9D                     LDX   DSCTMP
0329: 20 30 03                  JSR   XLIN1
032C: 4C 0D 03                  JMP   XP2
032F: 60            RET         RTS
0330: 48            XLIN1       PHA
0331: 38                        SEC
```

```
0332: E5 E0      56          SBC  $E0
0334: 48         57          PHA
0335: 8A         58          TXA
0336: E5 E1      59          SBC  $E1
0338: 85 D3      60          STA  $D3
033A: B0 0A      61          BCS  XLIN2
033C: 68         62          PLA
033D: 49 FF      63          EOR  #$FF
033F: 69 01      64          ADC  #$01
0341: 48         65          PHA
0342: A9 00      66          LDA  #$00
0344: E5 D3      67          SBC  $D3
0346: 85 D1      68  XLIN2   STA  $D1
0348: 85 D5      69          STA  $D5
034A: 68         70          PLA
034B: 85 D0      71          STA  $D0
034D: 85 D4      72          STA  $D4
034F: 68         73          PLA
0350: 85 E0      74          STA  $E0
0352: 86 E1      75          STX  $E1
0354: 98         76          TYA
0355: 18         77          CLC
0356: E5 E2      78          SBC  $E2
0358: 90 04      79          BCC  XLIN3
035A: 49 FF      80          EOR  #$FF
035C: 69 FE      81          ADC  #$FE
035E: 85 D2      82  XLIN3   STA  $D2
0360: 84 E2      83          STY  $E2
0362: 66 D3      84          ROR  $D3
0364: 38         85          SEC
0365: E5 D0      86          SBC  $D0
0367: AA         87          TAX
0368: A9 FF      88          LDA  #$FF
036A: E5 D1      89          SBC  $D1
036C: 85 1D      90          STA  $1D
036E: A4 E5      91          LDY  $E5
0370: B0 05      92          BCS  MOVEX2
0372: 0A         93  MOVEX   ASL
0373: 20 65 F4   94          JSR  LFTRT
0376: 38         95          SEC
0377: A5 D4      96  MOVEX2  LDA  $D4
0379: 65 D2      97          ADC  $D2
037B: 85 D4      98          STA  $D4
037D: A5 D5      99          LDA  $D5
037F: E9 00     100          SBC  #$00
0381: 85 D5     101  HCOUNT  STA  $D5
0383: B1 26     102          LDA  ($26),Y
0385: 45 30     103          EOR  $30
0387: 49 80     104          EOR  #$80
0389: 91 26     105          STA  ($26),Y
038B: E8        106          INX
038C: D0 04     107          BNE  XLIN4
038E: E6 1D     108          INC  $1D
0390: F0 14     109          BEQ  RET2
0392: A5 D3     110  XLIN4   LDA  $D3
0394: B0 DC     111          BCS  MOVEX
0396: 20 D3 F4  112          JSR  UPDOWN
0399: 18        113          CLC
039A: A5 D4     114          LDA  $D4
039C: 65 D0     115          ADC  $D0
039E: 85 D4     116          STA  $D4
03A0: A5 D5     117          LDA  $D5
03A2: 65 D1     118          ADC  $D1
03A4: 50 DB     119          BVC  HCOUNT
03A6: 60        120  RET2    RTS
03A7: 20 11 F4  121  HPLOT0  JSR  HPOSN
03AA: B1 26     122          LDA  ($26),Y
03AC: 45 30     123          EOR  $30
03AE: 49 80     124          EOR  #$80
03B0: 91 26     125          STA  ($26),Y
03B2: 60        126          RTS
--End assembly, 179 bytes, Errors: 0
```

ed me: it appeared that a modest 150 bytes or so could stand alone for XPLOT modification. I had to alter the argument-passing section and the code that stores to the screen memory, but of my final 179 bytes, about 80% is the original Applesoft HPLOT. It fits nicely in the free space starting at 768 ($300 hex), and can be called in all the ways familiar to HPLOT users. Assuming the assignment XP = 768, here are some examples of its use:

CALL XP,100,100 toggles the pixel at 100,100.
CALL XP,100,100 TO 200,100 plots a line in reverse mode from 100,100 to 200,100.
CALL XP, TO 100,100 plots from the last-plotted point to 100,100.
CALL XP, 0,0 TO 200,0 TO 100, 100 TO 0,0 plots a triangle in reverse mode.

Note that in this last example, the vertex 0,0 will be reversed twice, so a 'hole' will result. To close the curve, the point 0,0 can be XPLOTted a third time with:

CALL XP,0,0.

The general trick is to reverse each point in the shape you are drawing an odd number of times; this consideration is familiar to frequent XDRAW users.

Two final notes:

1. Since XPLOT simply reverses the background, no HCOLOR needs to be specified.
2. The XPLOT routine as it stands is not relocatable, so it should be BLOADed at address 768. Of course, if you have an assembler (preferably Big Mac), you can change the ORG statement to the location of your choice.

Happy Xplotting!

Bibliography.

1. *John Crossley, "Applesoft Internal Entry Points," Call -A.P.P.L.E. In Depth Number One: All About Applesoft, 1981, pp. 51-57. (Reprinted from the Apple Orchard)*
2. *Glen Bredon "FP Source" (Applesoft disassembly) Southwestern Data Systems*

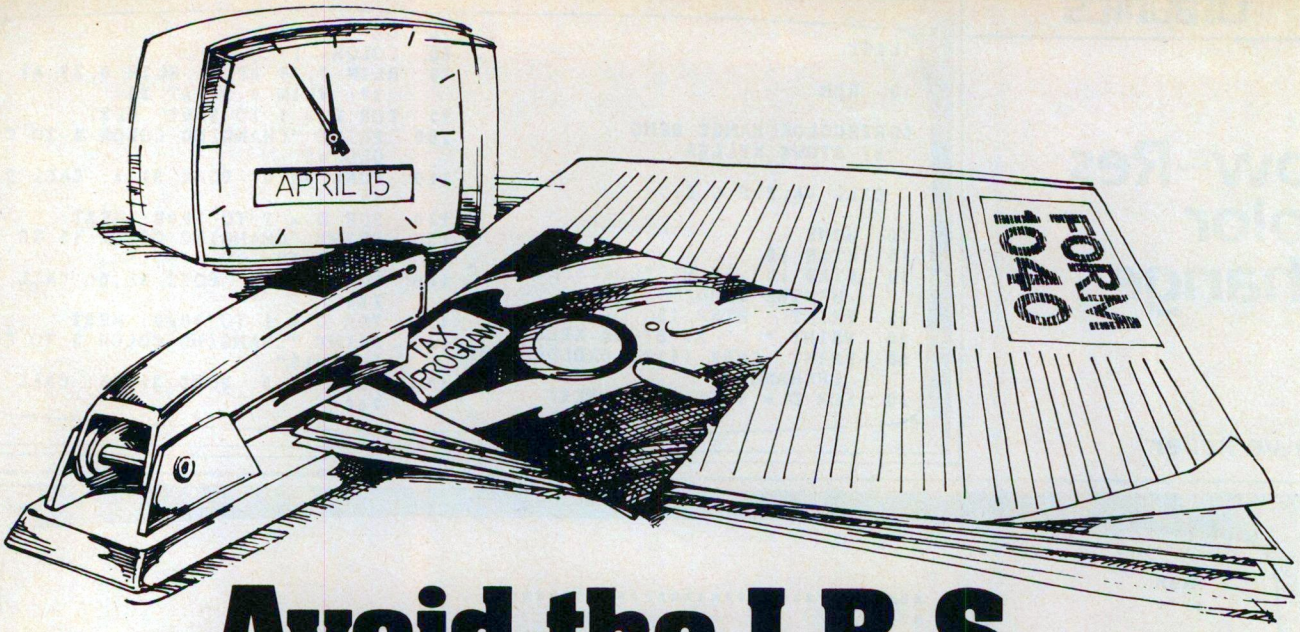
continued on p. 24

Low-Res Color Change

stowe keller

```
]LIST

10  REM

LORESCOLORCHANGE DEMO
  BY STOWE KELLER

  CALL -A.P.P.L.E.

20  HOME
25  HTAB 10
30  VTAB 10: PRINT "LORES COLOR C
     HANGER DEMO"
35  PRINT : HTAB 15
40  PRINT "     BY STOWE KELLER"
50  PRINT  CHR$ (4)"BLOADLORESCOL
     ORCHANGE"
60  FOR D = 1 TO 3000: NEXT
70  GR
80  COLOR= 15
90  HLIN 0,39 AT 0: HLIN 0,39 AT
     39: VLIN 0,39 AT 20
95  FOR D = 1 TO 3000: NEXT
100  PRINT "CHANGING COLOR 0 TO C
     OLOR=5"
110  POKE 30,0: POKE 31,5: CALL 7
     68
120  FOR D = 1 TO 3000: NEXT
130  PRINT "CHANGING COLOR 15 TO
     COLOR=0"
140  POKE 30,15: POKE 31,0: CALL
     768
150  FOR D = 1 TO 3000: NEXT
160  PRINT "CHANGING COLOR 5 TO C
     OLOR=15"
170  POKE 30,5: POKE 31,15: CALL
     768
```

```
:ASM
                        ********************************
                 2      *                              *
                 3      * LOW-RES COLOR CHANGE UTILITY *
                 4      *       by Stowe Keller        *
                 5      *                              *
                 6      *       Call -A.P.P.L.E.       *
                 7      *                              *
                 8      *                              *
                        ********************************
                10
                11      * Routine to convert all
                12      * occurrences of specified color
                13      * in Low-Res page to another
                14      * specified color.
                15      *   Assume that we are already
                16      * in Low-Res mode.
                17      *   From BASIC: POKE 30,[color to change from]
                18      *               POKE 31,[color to change to]
                19      *               CALL 768
                20
                21      *
                22      XCOORD    EQU  $CE          ;X-coordinate
                23      YCOORD    EQU  $CF          ;Y-coord
                24      FROMCLR   EQU  $1E          ;Color to change from
                25      TOCOLR    EQU  $1F          ;Color to chang to
                26      PLOT      EQU  $F800        ;Plot a point
                27      PLOT1     EQU  $F80E        ;Plot a point @ GBASL,H and mask; CLM=Y
                28      GBASCALC  EQU  $F847        ;Set GBASL,H from A*2
                29      SETCOL    EQU  $F864        ;Set Low-Res color
                30      SCRN      EQU  $F871        ;Get color of specific point @ (A),(Y)
                31                ORG  $300         ;For now
0300: A9 00     32      START     LDA  #0           ;Starting value
0302: 85 CF     33                STA  YCOORD
0304: A5 1F     34                LDA  TOCOLR       ;Get color to change to
0306: 20 64 F8  35                JSR  SETCOL       ;Set plotting color to desired color
0309: A9 00     36      XLOOP     LDA  #0
030B: 85 CE     37                STA  XCOORD
030D: A5 CF     38      LOOP      LDA  YCOORD       ;Get Y-coord
030F: A4 CE     39                LDY  XCOORD       ;(This can get confusing)
0311: 20 71 F8  40                JSR  SCRN         ;Get color of point on screen
0314: C5 1E     41                CMP  FROMCLR      ;Is it the same as color we want to change?
0316: D0 05     42                BNE  ENDXLP       ; No, loop
0318: A5 CF     43                LDA  YCOORD       ; Yes, so change it:
031A: 20 0E F8  44                JSR  PLOT1        ;Plot point with desired color
031D: E6 CE     45      ENDXLP    INC  XCOORD       ;Update Xcoord value
031F: A5 CE     46                LDA  XCOORD       ;See if we're at end of X loop
0321: C9 28     47                CMP  #40          ;Right edge of screen?
0323: D0 E8     48                BNE  LOOP         ; No, loop
                49      * We're at end of X-loop, now update Y-coord:
0325: E6 CF     50                INC  YCOORD
0327: A5 CF     51                LDA  YCOORD
0329: C9 28     52                CMP  #40          ;Bottom of screen?
032B: D0 DC     53                BNE  XLOOP        ; No, go do another horiz. line
032D: 60        54                RTS

--End assembly--

46 bytes

Errors: 0
```



Avoid the I.R.S.

(Inadvertently Ruined Software)

You need software insurance.

Diskettes are fragile, and when a protected program is damaged, the results are expensive and inconvenient. If you have a backup diskette, though, you can have your Apple, IBM or compatible computer back on line within seconds . . . affordably. That's software insurance.

Copy II Plus (Apple][,][Plus, //e)

This is the most widely used backup program for the Apple. Rated as "one of the best software buys of the year" by **InCider** magazine, its simple menu puts nearly every disk command at your fingertips. The manual, with more than 70 pages, describes protection schemes, and our **Backup Book™** lists simple instructions for backing up over 300 popular programs. A new version is now available that is easier to use and more powerful than before. Best of all, Copy II Plus is still only **$39.95.**

NEW!

WildCard 2 (Apple][,][Plus, //e)

Designed by us and produced by Eastside Software, WildCard 2 is the easiest-to-use, most reliable card available. Making backups of your total load software can be as easy as pressing the button, inserting a blank disk and hitting the return key twice. WildCard 2 copies 48K, 64K and 128K software, and, unlike other cards, is always ready to go. No preloading software into the card or special, preformatted diskettes are required. Your backups can be run with or without the card in place and can be transferred to hard disks. **$139.95** complete.

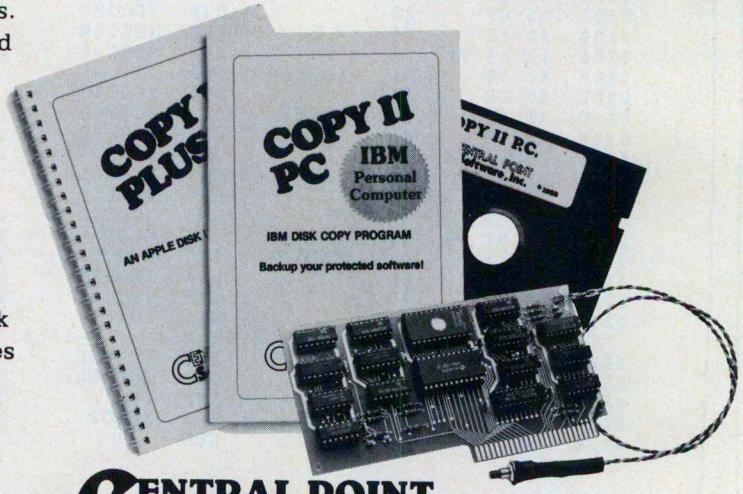
Important Notice: These products are provided for the purpose of enabling you to make archival copies only. Under the Copyright Law, you, as the owner of a computer program, are entitled to make a new copy for archival purposes only, and these products will enable you to do so.

These products are supplied for no other purpose and you are not permitted to utilize them for any use, other than that specified.

Copy II PC (IBM)

This is **THE** disk backup program for the IBM PC and PC/XT that backs up almost anything. Others may make similar claims, but in reality, nothing out performs Copy II PC . . . at any price. Copy II PC even includes a disk speed check and is another "best buy" at only **$39.95.**

We are the backup professionals. Instead of diluting our efforts in creating a wide variety of programs, we specialize in offering the very best in backup products. So, protect your software investment, **before** the I.R.S. gets you.



CENTRAL POINT Software, Inc.

The Backup Professionals

To order, call 503/244-5782, 8:00-5:30 Mon.-Fri., or send your order to: Central Point Software, 9700 SW Capitol Hwy, Suite 100, Portland, OR 97219. **Prepayment is required. Please include $2 for shipping and handling ($8** outside U.S. or Canada).

Circle 21 on Reader Service Card

THE SPREADSHEET SHEET:

Data Analysis

(PART 5)

jeffrey jacques / phd

Introduction

LAST time we noted how to statistically summarize the joint distribution of data when such data are categorized and analyzed with the use of contingency tables. This time we will explore further what has become the most widely used descriptive and predictive statistical measure — i.e., the linear equation. We have had some exposure to this equation last time when we discussed Pearsonian r computations. But here we will more thoroughly examine some of the essential concepts, how to compute the equation and its ancilliary statistics, and how to use it for effective data analysis.

Some General Thoughts on the Linear Model

The basic idea of linear regression is to fit a line so that it effectively summarizes the data, and thereby helps you to communicate your results to others. Essentially, every observed value of a data set, when summarized with a fitted line, can be divided into two components:

Observed value = fitted value + residuals.

The residuals are those values which should approach zero when we have a good fit between the observed values and the fitted line. The larger the values of the residuals, the poorer the summary provided by the fitted line and the poorer our predictive accuracy. The linear regression equation is computed on the basis of minimizing the sum of the squared residuals. More on this later. Now, let's examine the mechanics of linear regression.

The basic model of a straight (linear regression) line is:

Y = intercept pt. + slope * Xi + e

The slope simply reports the ratio of the change in the variable of interest (Y) to the change in the explanatory variable (X). The value of the slope may be computed from the principle of minimizing the sum of squares about the regression line with the following formula:

$$\text{slope} = \frac{\text{Sum (Xi - MeanX) * (Yi - MeanY)}}{\text{Sum(Xi - MeanX)}^2}$$

and, the

intercept pt = MeanY - slope * MeanX

The intercept point is that reference point where X equals zero and the regression line would cross the Y axis. From this equation and supplemental computations, we can determine:

1) whether or not a linear relationship exists between the variables,
2) the direction (negatively or positively associated) of the relationship,
3) the relative change in the variable of interest and the concomitant change in the predictive variable,
4) the predicted value of the variable of interest from any value of the explanatory variable, and
5) an estimate of the degree of accuracy of the predicted values.

All this, and more, can be yours by using your *Spreadsheet* software.

Modifying the Worksheet for Linear Regression

There are a substantial number of algebraically equivalent equations for computing the regression equation coefficients. Given the power of *THE Spreadsheet,* we are free to continue to use the conceptual formulae as the basis of our computational strategy. We will assign, therefore, a column for each of the following:

1) reproducing the explanatory variable (X), (in column H),
2) reproducing the variable of interest (Y), (in column I),
3) finding the sum of the squared deviations about the mean value for the predictive variable (X - MeanX)^2, (column J),
4) finding the sum of the squared deviations about the mean value for the variable of interest, (Y - MeanY)^2, (column K), and
5) finding the cross product deviations or the value of covariation — the numerator of the equation for the slope (column L).

We will: re-calculate the mean of each variable (cells H32 and I32), re-calculate the variance and standard deviations of each variable (cells J34, J35 and K34, K35, respectively), and summate the cross-product deviation values (cell L31). This process may appear redundant. That's because it is. But, it is also useful to replicate the original values so that you may, if you wish, transform the values (taking a log of a value, raising the values to a power, etc.) and always have easy access to the original values. This enables a variety of computational strategies in an efficient manner.

From these summated and computed values, we may determine quite easily the coefficients of the simple linear equation — i.e., the slope with the formula: L31/J31 and the intercept point with the formula +I32-(H32*I37).

Furthermore, once we have determined the slope, we can compute the Pearson product moment correlation

coefficient (r) by taking the ratio of the standard deviations and multiplying by the value of the slope. The coefficient of determination is not far away. We can easily compute it with a simple squaring of the value of r. (We have, therefore, one estimate of the accuracy of the regression equation. The RA2 value indicates our degree of reduction in error when we wish to predict the variable of interest from knowledge of the regression equation rather than using the mean score of the variable of interest.) Table 5.1 summarizes all the essential equations for each cell. Take a few minutes and study it.

By now you have noticed several additional elements of the worksheet. As we noted earlier, regression equation predictive precision can be estimated from knowledge of the residuals. The residuals are simply the difference between what was predicted and the observed values. We can compute a residual for each Y value with the formula:

Residual = Yi - (intercept + slope * Xi)

By squaring this value and then summating all the squared residual values (column M), we can estimate the residual variance (in cell M41 enter +M31/(M28-2)), and the standard deviation of the regression line (in cell M42 enter @SQRT(M41)). Ideally, this value should be very small, thus indicating a clustering of values near or on the regression equation.

Go ahead and enter these formulae into your worksheet. Use the faculty data set, too. I'll wait....

Further Thoughts on Constructing Linear Regression Worksheets

Even in a small study, the number of different regression equations and the summary correlation coeffi-

	A	B	C	D	E	F	G	H	I	J	K	L	M
1			Faculty Data Set -- 1983							Scratch	Space		
2													
3	Faculty ID	Rank	Degree	Income	Yrsemp	Sex		X	Y	(X-M)^2	(Y-M)^2	x * y	Resid^2
4													
5	Faculty # 1	3	3	18442	8	1		3	18442	(H5-H32)^2		(H5-H32)*(I5-I32)	
6	Faculty # 2	1	2	12167	7	0		1	12167				
7	Faculty # 3	3	3	16858	19	0		3	16858		(I7-I32)^2		(I7-(L38+(L37*H7)))^2
8	Faculty # 4	3	3	20641	1	1		3	20641				
9	Faculty # 5	2	2	25000	1	1		2	25000				
10	Faculty # 6	4	3	20970	23	1		4	20970				
11	Faculty # 7	1	1	14775	1	1		1	14775				
12	Faculty # 8	2	2	15441	20	0		2	15441				
13	Faculty # 9	2	2	15595	17	0		2	15595				
14	Faculty #10	2	3	17940	8	0		2	17940				
15	Faculty #11	1	1	9525	1	0		1	9525				
16	Faculty #12	4	3	20690	11	0		4	20690				
17	Faculty #13	1	2	11865	1	1		1	11865				
18	Faculty #14	2	2	14381	31	0		2	14381				
19	Faculty #15	1	1	8250	0	1		1	8250				
20	Faculty #16	2	2	15000	4	1		2	15000				
21	Faculty #17	3	3	17117	7	0		3	17117				
22	Faculty #18	4	3	28414	2	1		4	28414	(H22-H32)^2		(H22-H32)*(I22-I32)	
23	Faculty #19	3	3	17424	5	1		3	17424				
24	Faculty #20	1	1	8494	3	0		1	8494		(I24-I32)^2		(I24-(L38+(L37*H24)))^2
25													
26	Summary	Statistics											
27													
28	Count	20	20	20	20	20		@COUNT(H5...H24)					@COUNT(H5...H24)
29	Minimum	1	1	8250	0	0							
30	Maximum	4	3	28414	31	1							
31	Sum	45	45	328989	170	10		@SUM(H5...H24)					@SUM(H5...H24)
32	Average	2.25	2.25	16449.	8.5	.5		@AVE(H5.H23)		@AVE(I5.I24)			
33	Range	3	2	20164	31	1							
34	Variance	1.1447	.61842	2.68E7	79	.26316				+J31/(J28-1)	+K31/(K28-1)		
35	St. Dev.	1.0699	.7864	5174.8	8.8882	.51299				@SQRT(J34)	@SQRT(K34)		
36	Skewness	.30439	-.496	.4274	1.2028	0							
37	Kertosis	-1.092	-1.152	.34442	.62441	-2.235					slope = +L31/J31		
38											interc= +I32-(H32*L37)		
39											r = +J35/K35*L37		
40											RA2 = (J35/K35*L37)^2		
41											Res.Vr=		+M31/(M28-2)
42											St.Err=		@SQRT(M41)
43							G	H	I	J	K	L	
44								Bivariate Correlation Summary					
45													
46							Dep.Vr	Ind.Vr	Intcpt	Slope	RA2	St. Err	
47													
48							Income	Rank					
49							Income	Degree					
50							Income	Sex					
51							Income	Yrsemp					
52							Rank	Degree					
53							Rank	Yrsemp					
54							Rank	Sex					
55													

NOTE: Columns do not perfectly align so that you may more easily see the appropriate formulae.

Table 5.1. Worksheet and calculations for simple regression.

cients we could produce is quite large. For example, in this study, if we treated each variable as a possible variable of interest and as a predictor, we would have 20 different equations. Why 20? Well, since the equation of Y on X (e.g., income on degree) will generally not equal X on Y (e.g., degree on income), the number of equations is equal to:

number of variables * (number of variables - 1)

With five variables, there are, probably, 20 different equations. It is necessary, therefore, to further identify your variable(s) of interest and the explanatory variables. By identifying which variables we are trying to explain, we can substantially reduce the number of equations. Easy to say, but more difficult to do.

In this study, we have been interested in the reward system for college faculty. The most important variable has been that of income. We will, therefore, regress income on each of the other variables. (That's right. Whichever variable is stated first is the variable of interest and then the predictor variable is stated.) Many would argue that faculty rank is an additional component of the reward system. Not only is rank an important predictor of income, but, it is also a variable of interest and can be estimated from the remaining explanatory variables.

Using just income and rank reduces the number of equations to 7 rather than 20. In the lower right side of Table 5.1, I have developed a table which will handle the necessary information and the most important computed values. (Columns G through L, rows 43 to 55 are used for this purpose). Go ahead and complete your worksheet. Now, go ahead and do the necessary computations. Then,

	A	B	C	D	E	F	G	H	I	J	K	L	M
1				Faculty Data Set -- 1983						Scratch	Space		
2													
3	Faculty ID	Rank	Degree	Income	Yrsemp	Sex		X	Y	(X-M)^2	(Y-M)^2	x * y	Resid^2
4													
5	Faculty # 1	3	3	18442	8	1		3	18442	.562500	3.97E6	1494.4	774570
6	Faculty # 2	1	2	12167	7	0		1	12167	1.56250	1.83E7	5353.1	255322
7	Faculty # 3	3	3	16858	19	0		3	16858	.562500	166913	306.41	6.07E6
8	Faculty # 4	3	3	20641	1	1		3	20641	.562500	1.76E7	3143.7	1.74E6
9	Faculty # 5	2	2	25000	1	1		2	25000	.062500	7.31E7	-2138.	9.04E7
10	Faculty # 6	4	3	20970	23	1		4	20970	3.06250	2.04E7	7911.0	4.76E6
11	Faculty # 7	1	1	14775	1	1		1	14775	1.56250	2.80E6	2093.1	9.69E6
12	Faculty # 8	2	2	15441	20	0		2	15441	.062500	1.06E6	252.11	2590.9
13	Faculty # 9	2	2	15595	17	0		2	15595	.062500	730085	213.61	10629.
14	Faculty #10	2	3	17940	8	0		2	17940	.062500	2.22E6	-372.6	5.99E6
15	Faculty #11	1	1	9525	1	0		1	9525	1.56250	4.79E7	8655.6	4.57E6
16	Faculty #12	4	3	20690	11	0		4	20690	3.06250	1.80E7	7421.0	6.06E6
17	Faculty #13	1	2	11865	1	1		1	11865	1.56250	2.10E7	5730.6	41329.
18	Faculty #14	2	2	14381	31	0		2	14381	.062500	4.28E6	517.11	1.23E6
19	Faculty #15	1	1	8250	0	1		1	8250	1.56250	6.72E7	10249.	1.16E7
20	Faculty #16	2	2	15000	4	1		2	15000	.062500	2.10E6	362.36	241967
21	Faculty #17	3	3	17117	7	0		3	17117	.562500	465623	500.66	4.86E6
22	Faculty #18	4	3	28414	2	1		4	28414	3.06250	1.43E8	20938.	2.77E7
23	Faculty #19	3	3	17424	5	1		3	17424	.562500	949748	730.91	3.60E6
24	Faculty #20	1	1	8494	3	0		1	8494	1.56250	6.33E7	9944.3	1.00E7
25													
26	Summary	Statistics											
27													
28	Count	20	20	20	20	20		20	20	20	20	20	20
29	Minimum	1	1	8250	0	0							
30	Maximum	4	3	28414	31	1							
31	Sum	45	45	328989	170	10		45	328989	21.7500	5.09E8	83307.	1.90E8
32	Average	2.25	2.25	16449.	8.5	.5		2.25	16449.5				
33	Range	3	2	20164	31	1							
34	Variance	1.1447	.61842	2.68E7	79	.26316				1.14474	2.68E7		
35	St. Dev.	1.0699	.7864	5174.8	8.8882	.51299				1.06992	5174.8		
36	Skewness	.30439	-.496	.4274	1.2028	0							
37	Kertosis	-1.092	-1.152	.34358	.62441	-2.235							
38											slope =	3830.2	
39											interc=	7831.5	
40											r =	.79194	
41											R^2 =	.62718	
42											Res.Vr=		1.05E7
43											St.Err=		3246.2
44													
45													
46													
47													
48													
49													
50													
51													
52													
53													
54													
55													

Bivariate Correlation Summary

Dep.Vr	Ind.Vr	Intcpt	Slope	R^2	St. Err
Income	Rank	7831.5	3830.2	.62718	3246.2
Income	Degree	5878.3	4698.3	.5098	3722.2
Income	Sex	14821	3257.3	.104	5031.6
Income	Yrsemp	16105	40.5	.005	5303.5
Rank	Degree	-.38298	1.17	.73979	.5607
Rank	Yrsemp	1.93	.038	.0978	1.044
Rank	Sex	2.1	.3	.021	1.088

Table 5.2. Data values for simple regression.

check your values with those values listed in Table 5.2. No rush, I'll wait for you. . . .

Using the Summarized Values

Okay, now that we have computed all these values, what do they mean and how can I use them? All right. For this data set, the best predictor of income can be determined by examining the RΛ2 values. Clearly, the variable of rank was the best predictor of income. But, since many would argue that income and rank are part of the same reward system, the variable of degree is probably the best independent predictor of income (even though it has a smaller RΛ2 value). Why? Well, the degree(s) faculty earn are generally not under the control of the employing college. It is, therefore, a truer independent measure for predicting income. Sex was of considerably less importance. Perhaps most interesting, for this data set at least, the number of years employed had very little to do with the salary structure.

Returning to the relationship between degree and income, each advanced degree was worth (on the average) about $4700. We know this from the value of the slope — i.e., $ 4698.3. Similarly, since we coded females as 0 and males as 1, being male was worth about $3,000. But, before you run to cry sexual bias, the small reduction in error (RΛ2 value) and the large standard error of the estimate ($ 5,031.6) indicates a great deal of inaccuracy in this $3,000 difference between the sexes. (We'll talk more about our level of faith in the equation in a later article when we discuss inferential statistics.)

Predictives

As we indicated earlier, one of the great advantages of this equation is the ability to use it as a predictive tool. Where your RΛ2 is large and the standard error is small, prediction of the variable of interest from the intercept and slope values is quite accurate and quite easy. Just extend your summary table. Reserve column M for the predictive (e.g., rank or degree) value and use column N for the predicted (e.g., income) value. In cell N48 enter:

+ I48 + (J48 * M48)

/Reproduce this equation through N54. You are now ready to enter the appropriate predictor values in column M. Row 48 will predict salary from rank, row 49 will predict salary from degree, etc. All you need do is to press the ! key. Pretty easy and pretty neat!

Conclusion

With this article, we have now completed our review of descriptive statistical summary procedures. Next time we will explore how much faith we may have in these measures when we wish to infer from our random sample back to the population, that is, do these summary measures indicate that a relationship exists between the variables in the population? We will begin our examination of the world of inferential statistics. And, for each statistical test, we shall construct a generalized template which you can use in a wide variety of applications. See you then.

DISK DRIVES $199

First time under $200!

At last you can afford to buy another drive!!

- Siemens drive mechanism (uses lead screw instead of rotating cam, for very quiet, reliable operation)
- Works with your present controller
- Can be used as 35 or 40 track drive
- Improved self-centering of diskettes
- Externally-adjustable speed control
- Handles synchronized tracks
- Manufacturer's 1-year warrantee

These superb ready-to-use drives come with standard size and color casings. *Disk controller cards available: $65.00*

Add $5 for shipping and handling. Send money order or certified check for immediate shipment, $4 extra for 2nd day delivery
Ask for free 74LS125 chip with order.

VF ASSOCIATES
6327 Western Ave, N.W., Wash., D.C. 20015
(202) 363-1313

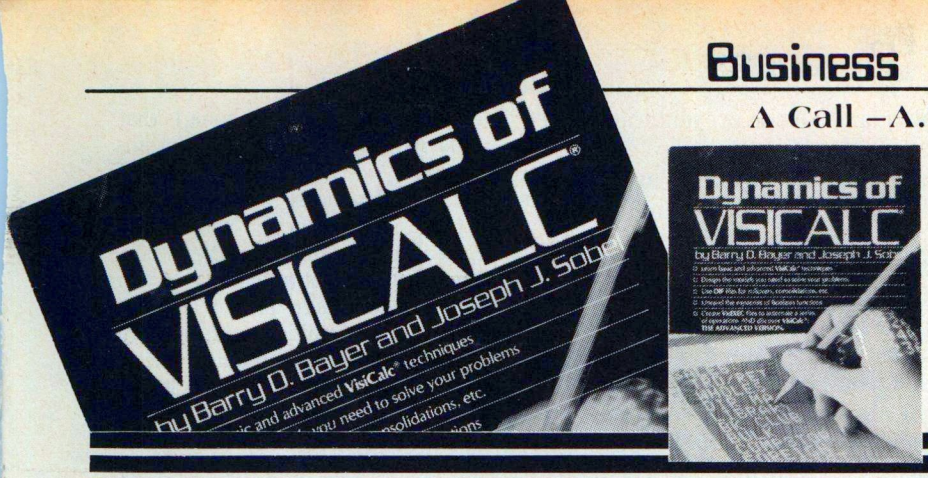
Circle 8 on Reader Service Card

XPLOT

from p. 18

```
]
5  REM

 XPLOT DEMO  BY MARK
      HARRIS

CALL -A.P.P.L.E. : APR.
      1984

10  HOME
15  PRINT  CHR$ (4)"BLOAD
      XPLOT"
20  HGR : HCOLOR= 3
30  FOR Y = 0 TO 159
      STEP 7: HPLOT
      0,Y TO 279,Y:
      NEXT
40 XP = 768: VTAB 23:
      PRINT "'S' TO
      STOP, ANY OTHER
      KEY TO TOGGLE";
50  GOSUB 100: GET A$:
      ON A$ <  > "S"
      GOTO 50
60  TEXT : HOME : END

100  CALL XP,25,25:
      CALL XP,25,25
      TO 125,25 TO 125,125
      TO 25,125 TO 25,25:
      RETURN
```

A Call –A.P.P.L.E. BOOK REVIEW:



Dynamics of VisiCalc

jeffrey jacques/phd

DYNAMICS OF VISICALC $19.95
Barry Bayer and Josepf Sobel
Homewood, Illinois: Dow Jones–Irwin.
1983. (800) 323-1719

THIS is one book in a growing series of books about one of the most important micro–computer packages available today — electronic spreadsheet programs. Bayer and Sobel promise the reader an opportunity to:

1. learn basic and advanced VisiCalc™ techniques,
2. design models to solve your spreadsheet problems,
3. use DIF ™ files for rollovers, consolidations, etc.,
4. unravel the mysteries of Boolean functions,
5. create VisExec files to automate a series of operations, and
6. discover VisiCalc: The Advanced Version.

This is a tall order for even the best of books. Unfortunately, this is not one of the best of books. Bayer and Sobel are successful with some of these goals, but fail with others.

Most Calc books seem to fall along a multidimensional continua. Some books present tutorial information on how to load such software, understand the structure of the worksheet, and use the commands and function of VisiCalc. Books like Castlewitz's, *The VisiCalc Program Made Easy* is a typical example of this approach. At the other extreme are those books that present templates and little or no information in a tutorial format. Such books as Flast's, *54 VisiCalc Models,* and Williams, Taylor, and King's, *The Power of VisiCalc,* Volumes 1 and 2, are examples of the latter. *The Dynamics of VisiCalc* falls more toward the tutorial end of the continua than the template/model end. Indeed, Bayer and Sobel note that their book is not designed for those trying to find completed models, templates or grids. Rather, they argue, by merging your knowledge of VisiCalc with your own unique needs, you can more easily develop appropriate Calc models than you could by modifying and altering existing templates.

If you have considerable experience with VisiCalc, or other Calc programs, this may be true. I believe for most people, however, organized models and well developed templates are a useful step in seeing how others solve particular problems, and, thereby, help in the solution of your own problems. Because these authors reject the need to present well developed templates, very few models are explored and all too few good illustrations and application are given.

For those of you who already own a computer and have purchased a Calc program, Bayer and Sobel recommend that you work through the Calc manual. I agree. The tutorial in most Calc books is worth the effort. Similarly, they have other good recommendations, including:

FOR APPLE II PLUS, FRANKLIN, APPLE IIe

Font Down Loader

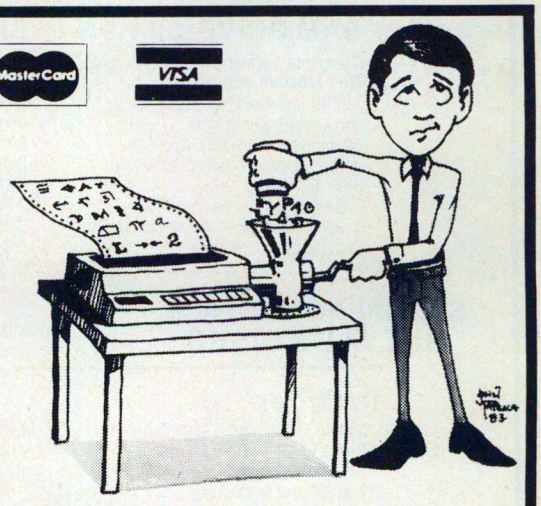
Expand the capacity of your printer hundreds of times

Load custom fonts into your Apple® Matrix Printer, Prowriter™ 8510A, OKI® Microline 92, 93, 84 Step II, and Epson® FX and use them with virtually every word processor to turn your printer into a **custom typesetter.** After the fonts are loaded, they will stay in your printer until it's turned off. A font editor is also provided to allow you to create your own graphics, text, foreign language letters, math and electronics symbols to load into your printer. On-Disk (Specify Printer)

. . . . . $39.00

New improved versions with drivers for Grappler, Pkaso, Wizard and most other intelligent parallel boards.

MasterCard VISA



"There's got to be a better way to load fonts!"

$100 REWARD

Submit the best or most unique font using the above software and we will make you $100 richer. Other prizes for the first 25 runners up.

MICRO WARE
P.O. Box 113
Pompton Plains, N.J.
07444

Dealer and Distributor Inquiries Invited

CALL (201) 838-9027

Circle 9 on Reader Service Card

1. back up your diskettes,
2. build templates using both basic and Advanced VisiCalc techniques — where possible, and
3. avoid forward referencing of cells.

If all you have is the original version of VisiCalc — with its memory limitations and problems of formatting output, then you may find the extensive write-up about using overlays, VersaCalc 16, and VisExec useful. If you have THE Spreadsheet, MagiCalc, or some other version, then you may not find it as useful. Information about "pre-boot programs" which may be used to increase memory is not very useful for those who own THE Spreadsheet, which requires no pre-boot.

I was looking forward to a wealth of information and applications on Boolean functions; given that the authors promised to "unravel the mysteries of Boolean Functions." The authors present basic information, several truth tables, and a couple of examples on using the @IF statement. The hyperbole of "unraveling the mysteries" does not match the quality of the presentation. I was disappointed.

The authors' best work is in the area of financial model template building — Chapter 6. They seem to be very comfortable taking you through the pitfalls of preparing a projected income statement, cash flow analysis, and balance sheet. If this is what you need, and if you are using VisiCalc — in either its original or Advanced version, then the purchase price of $19.95 may be worth it.

Publishers Comments

Thank you for the galley of Jeffrey Jacques' review of *Dynamics of VisiCalc.* It is obvious that Mr. Jacques prefers books which are closer to the tutorial end rather than the template/model end of his one dimensional continuum. Dynamics, of course, is on the other end. It would have been no problem at all to present a dozen, or 54, templates to calculate gas mileage, prepare a home budget, chart your stock market successes or shine your shoes. Indeed, we recognize that there is a place for such books. What we have tried to do in Dynamics, however, is a bit different.

The major premise of the book is that VisiCalc is, in effect, a powerful programming language with a very small set of commands and functions, and a very rigid syntax. But this language (or Magicalc or THE Spreadsheet, which for these purposes constitute the same language) can be learned by computer users who have no intention of becoming real programmers. (In other words, people like us.)

Such people, if they haven't purchased a computer yet, are currently solving their problems using pencil and hand calculator, and know exactly what must be done to their data to accomplish the desired result, even though they could never express such ideas in a program in BASIC or Pascal. It is our hope that Dynamics of VisiCalc will assist such users.

As noted by Mr. Jacques, Dynamics contains only one large template; what he has not mentioned are the many small examples showing how to attack various common problems using a Calc program. For example, very little has been written on consolidating and accumulating, two extremely important business uses. (Multiplan's major claim to fame, of course, is its automatic spreadsheet "linking" ability, but neither VisiCalc nor Magicalc do the job automatically.) In our book, we illustrate how to accomplish these tasks using either data interchange format (DIF) or "PRINT TO DISK" Visexec files in ways that have not been documented by the calc program manufacturers. These and other functions are every bit as important for THE Spreadsheet users, as for VisiCalc users.

Mr. Jacques accuses us of not delivering on the "Hyperbole" of "Unraveling the Mysteries" of the Boolean functions, citing our inclusion of some basic information, several truth tables and a couple of &IF examples. Yes, the Booleans are mysterious to many; but as Mr. Jacques and many of you and ourselves know, Booleans are not very complicated once you understand the basic definitions of &AND and &OR and &NOT and the rest. Mr. Jacques obviously doesn't need any help, but our "Hyperbole," if that it be, was addressed to the calc user who has never been exposed to the basic definitions, via truth table or otherwise, i.e., to those for whom Booleans are mysterious.

Noting Mr. Jacques's silence as to our success with the other topics mentioned on the book cover, we assume that he thought we delivered on those "Promises." Only the reader can judge whether the Dynamics of VisiCalc is worth $19.95. To assist in that decision, our publisher, Dow Jones — Irwin, will sell Dynamics on a direct basis with a 15 day money back guarantee.

. . . Barry D. Bayer
. . . Joseph J. Sobel

ONE PASS COPY **$29.95**

The Copy Machine
Does for disks what Xerox did for paper.

- Don't let backing-up slow you down.
- Examples:
  Copy Apple System Master on one drive in one pass and 38 seconds compared to two and a half minutes with COPYA.
- **Copy a disk with a 128 sector game on two drives in EIGHTEEN SECONDS (!) vs. 1 minute 33 seconds with COPYA.**
- This is the fastest possible copy system on Apples.
- Make backing up a pleasure. Copy most disks in one pass and a fraction of the time.
- Change parameters — INIT, bypass bad sectors, etc.
- If you backup your work, you owe it to yourself to use ONE PASS COPY.

RAM DRIVE $24.95

- Use your extra memory as a Disk Drive. **No hardware needed.**
- All DOS commands work the same.
- 310 sectors with a 128K Apple IIe, 63 sectors on any 64K Apple IIe or II+. That's more room than any other software.
- Incredibly fast — you have to see it to appreciate the speed and reliability.
- Full package of utilities.

SPEED-DOS $24.95

- Improves SAVE, LOAD, BLOAD, BSAVE, RUN and BRUN times **up to 500%.**
- Compatible with RAM DRIVE, all DOS commands, most programs.
- Bload HI-RES screen from floppy in 3 seconds, from RAM DRIVE in half a second!
- Completely unprotected. Add to any program.

SPECIAL POWER PACK
GET ALL 3 DISKS for only **$49.95**

To Order:

1) Mail or phone orders accepted.
2) Check, COD, VISA or MC (include exp. date and signature)
3) Add $1.50 U.S. shipping
4) Add $5.00 foreign
5) Specify 64K or 128K Apple when ordering RAM DRIVE

- Easy to use documentation.
- All disks copyable, catalogable.
- Dealer inquiries welcome.

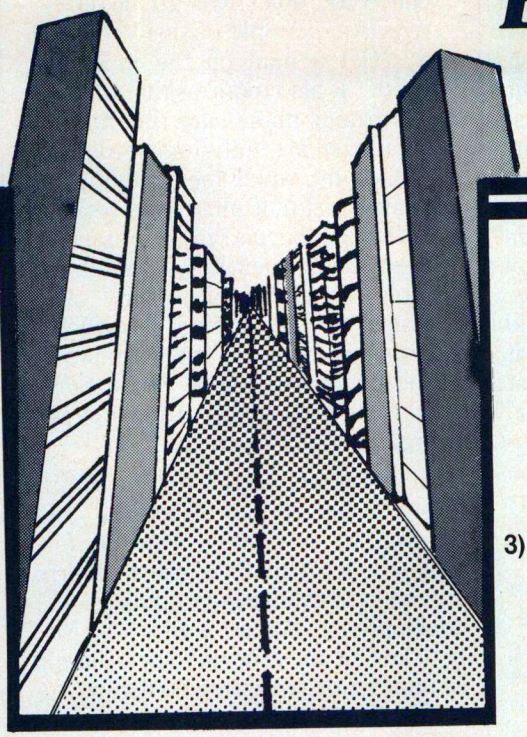
4/84

Software banc, inc.
1225 N. Water Street
Milwaukee, WI 53202
(414) 271-0100 (312) 876-0715

Circle 10 on Reader Service Card

Strictly Business

mike regeimbal



ORGANIZING is the second broad area of management as identified by Harold Kontz and Cyril O'Donnell in their book *Principles of Management.* Of the five areas (planning, organizing, staffing, directing, and controlling), organizing has the distinction of being around the longest. Eight hundred years ago, the Catholic Church brought forth one of the first organization structures by identifying in Cannon Law the responsibilities and prerogatives of the church hierarchy.

Organization structures should not just evolve. To favor productivity improvement, organization structures should be the reflection of a scheme, a plan which has incorporated sound rational thought. The organization structure should complement the objectives to be met.

Peter Drucker, in his book *Management, Tasks, Responsibilities, and Practice* identifies five different organization structures.

1) Functionalism

This type of organization structure was first espoused by Henri Fayol, a French industrialist around 1910. A functional design consists of organizing work into related bundles of skill. The work moves while the position of the worker is fixed. Functional design provides clarity of personal task while making it difficult to understand the task of the whole. It does very poorly at developing and preparing the managers of tomorrow (the pigeon hole effect).

2) Team

This structure consists of workers with different backgrounds, skills, and knowledge, drawn from various organizations working together on a specific task. Here the workers move while the position of the work is fixed. In a team environment, everyone knows the work of the whole and holds himself responsible for it. This approach is highly receptive to new ideas and adaptations. The negative aspect of the team approach is that it is less productive because it demands continuous deliberation, explanation, and communication. A large percentage of members spend their time simply keeping things running.

3) Federal Decentralization

This type of organization structure became popular when it was implemented at General Motors in 1921. Implemented by Alfred P. Sloan, it became the prototype for decentralization. Sloan used Fayol's functional structure for the subunits while organizing the company itself on the basis of decentralized authority with centralized control. As an example, at G.M. each of the automotive divisions was responsible for its own design, engineering, manufacturing, marketing, and sales. A division's profit and loss became part of the company's profit and loss. Federal decentralization provides clarity of task and economy. It makes it easy for each member of the autonomous business to understand his own task and the task of the whole business. The greatest strength of this principle is with respect to manager development. This is because it prepares and tests people for top management responsibility at an early stage. Its basic limitation is application. The majority of companies find it difficult to organize into separate autonomous "businesses."

4) Simulated Decentralization

Based upon the premise that some companies do not lend themselves to federal decentralization, this organization structure is a second best alternative. Simulated decentralization espouses setting up units with the maximum possible autonomy. This would include a unit's own management and at

BIZPAK II: Manufacturer's Comments

BEFORE bringing BIZPAK II to market, we asked a great number of owners of small businesses for their ideas and suggestions. Several versions of the program were then prepared to test these ideas.

It was about that time that I sent you a pre-publication edition of BIZPAK II. Now it turns out that whoever packed the shipping carton grabbed a mimeographed manual with one of those harebrained ideas that didn't fly. It is no wonder that Mike Regeimbal on page 34 of the February issue of *Call-A.P.P.L.E.* put BIZPAK II on the carpet. Really, what else could he do! Fortunately for us, the real BIZPAK does not suffer from the faults he described. Unfortunately — now it is in print in a prestigious magazine.

One point I failed to make in my previous letter, something which comes up quite often: How about hard-disk compatibility? Well, it is unnecessary since our operating system functions similar to a RAM disk. There is hardly any disk drive activity while the system is running.

Horst K. Schneider
Mid-West Marketing
1492 Ammons — Suite 8
Denver, CO 80215

least a simulation of profit and loss responsibility. These units would then buy and sell to each other, using transfer prices determined internally. IBM is the leading example of simulated decentralization. It has split its computer business into two key units. Marketing and servicing make up one unit while development and manufacturing make up the other. The problem with this method of organization structure is that it rarely satisfies the specification: Everyone should know his own task. In addition, it does not satisfy the demand that managers should be able to understand the job of the whole. Simulated decentralization should be viewed as a means of last resort for organization structure.

5) Systems Structure

This organization structure is an extension of the team principle. However, instead of individuals comprising a team, the team is made up of a wide variety of different organizations. Systems structure first became visible when implemented by NASA in its organization of the U.S. space program in the 1960's. This organizational structure demands absolute clarity of objectives to be effective. An example of this would be to "Get a man on the moon by 1970." It also requires that managing groups make sure the mission objectives are fully understood by everybody, which means that each managerial unit must take responsibility far beyond his own assignment. Consequently systems structures usually lack clarity and stability. People find it difficult knowing exactly what their job is and understanding its relationship to the whole under this type of organization structure.

If the above is the extent of different organization structures and criteria for evaluation can be formulated, we can then use a matrix approach to evaluating organizational development. Again, we turn to Mr. Drucker for the criteria to be used for evaluating organization structure.

1) Clarity

Each managerial component and each individual within the organization, especially the manager, needs to know where he belongs, where he stands, and where he has to go for whatever is needed, whether for information, cooperation, or decision. A structure in which no one knows without an elaborate organizational manual where he belongs, where he has to go, and where he stands, creates friction, wastes time, causes bickering and frustration, delays decisions, and is altogether an impediment, rather than a help to the organization.

2) Economy

The minimum effort should be needed to control, to supervise, and to coax people to perform. The organization structure should encourage self control and self motivation.

3) Direction of Vision

The organization structure should direct the vision of individuals and managerial units toward performance rather than toward efforts. It must not make effort more important than result. It should not make craftsmanship an end in itself. It must, in brief, make willingness and ability to work for results rather than for the sake of work itself, to work for the future rather than rest on the achievements of the past, and to strive for strength rather than to put on fat.

4) Understanding Personal and Common Task

An organization should enable each individual, as well as each managerial component, to understand his own task. At the same time, an organization should enable everyone to understand the common task, the task of the entire organization. Each member of the organization, in order to relate his efforts to the common good, needs to understand how his task fits in with the task of the whole, and in turn, what the task of the whole implies for his own task, his own contributions, his own direction.

5) Decision Making

Decisions have to be made, made on the right issues and at the right level, and have to be converted into work and accomplishment. An organization design, therefore needs to be tested as to whether it impedes or strengthens the direction of the decision making process.

6) Stability and Adaptability

An organization must be able to do its work even though the world around it is in turmoil. It must be able to build on its performance and achievement of yesterday. It needs to be able to plan for its own future and continuity. But stability is not rigidity. A totally rigid structure is not stable, it is brittle. Only if the structure can adapt itself to new situations, new demands, new conditions, new faces and personalities, will it be able to survive. Adaptability is a major requirement therefore.

7) Perpetuation and Self renewal

An organization must be capable of producing from within tomorrow's leaders. It must have the ability to test a man on each level for the next level above. For perpetuation and self renewal, it is further necessary for an organization structure to be accessible to new ideas and to be willing and able to do new things.

With this information, you should be able to evaluate your own organization structure. As you can see, a lot of emphasis is placed on information availability and flow. It is possible that your computer will enhance information availability and thus give

FREE?DISKETTES

SAVE MONEY ! Apple II+/e users can use the diskette flip side, if another "write enable" notch is correctly made.

The DISK-NOTCHER by QUORUM quickly SOLVES that PROBLEM. It's like FREE DISKETTES!

- Stainless Steel Guide
- Easy Leverage Handle
- Clippings Catcher
- Square Notch Cut
- Black Finish
- Get THE BEST!

Certifix™

BE SAFE ! Your 'FREE' disk is CERTIFIED 100% ERROR FREE with CERTIFIX by QUORUM. It 'LOCKS OUT' DISK FLAWS and lets you use the rest. Displays status report & saves it to disk. Next, CERTIFIX automatically formats then offers to initialize with genuine Apple DOS 3.3 too. Great for testing economy disks. CERTIFY, FIX & INITIALIZE every disk with CERTIFIX !

100% Money Back Satisfaction Guarantee !

DISK NOTCHER is $14.95
CERTIFIX™ is just $24.95
ONLY $29.95 for BOTH!
Add $1.50 s/h • CA add 6½ % tax

QUORUM INTERNATIONAL, Unltd.
INDUSTRIAL STATION P.O. BOX 2134-CA
OAKLAND, CA 94614

Circle 11 on Reader Service Card

you options previously not available. For example, after analyzing your particular situation you may find that one of the reasons you are deterred from the team (vs. functional) approach is the incessant demand for collecting and disseminating information. To eliminate this obstacle, you might want to implement a computer based memo/mail package.

To reiterate, organization structure should follow a design and not just happen. Only by incorporating the organization structure into the master plan scheme of things can optimum benefits of intelligent organization structuring be derived.

Book Review

This month's selection is entitled *The Personal Computer Book* by Peter A. McWilliams. The author also writes a syndicated column *On Line* you may have seen in your Sunday newspaper. I would wholeheartedly recommend this book. It is delightful reading. Mr. McWilliams has a unique ability to present "computereez" in a simple humorous manner.

The book is divided into three sections. The first section is a discourse on what computers are, how they work, and the part that programs play. The book starts with an explanation of some of the jargon, technical terms, and buzz words. This is followed by a review of the history of computers, starting with the abacus and continuing through to a comparison between the modern day PC and the first fully electronic computer, the ENIAC. The remainder of this section is about programming, and the information available through data banks.

The second section reviews the uses of the personal computer. The subjects covered include personal computer uses in business, education, and in the home. Separate chapters also address word processing, computer games, and the impact of the computer on children. There is also a chapter on what computers do not do well. He cites such activities as using a PC for an appointment calendar, preparing personal income tax returns, a recipe book, or balancing the check book as being impractical or too expensive for most PC users.

The last section of this book concerns selecting and purchasing a personal computer. Included is a name brand buyers guide to 28 of the more popular brands now on the market. His most enthusiastic review was given to the IBM PC because it receives good support (both from IBM and software houses) and its technology is state-of-the-art. About the Apple he indicates: "The Apple][and Apple *III* are, dollar for dollar, not very good values... with the Apple][, you are paying for a computer design that is six years old, ancient by personal computer standards, and of course, the ads necessary to keep it selling." As you can see he is not exactly an Apple fan.

I would especially recommend this book for the first time buyer or those people who would like to understand computing through the eyes of a humorous writer. In my opinion it is one of the more interesting and helpful computer books on the market today.

New DAVID-DOS II™ Compared With Other DOS's

Now you can speedup Text-file handling two ways

DAVID-DOS II is a new edition of DAVID-DOS with added speed, commands, and features. New Read, Write and Save routines are high speed. DAVID-DOS II updates full disks like Apple's Master Create. (The programs on your disks are not touched). DAVID-DOS II Inits blank disks with Basic, Binary or Exec HELLO in seconds. Ten new commands operate identical to existing DOS commands. Use them from the keyboard or in Basic programs. They accept A & L parameters.

Ten New DOS Commands

1. **TLOAD speed loads all Text Files, random or sequential, to ram.**
2. **TSAVE speed saves all Text Files, random or sequential, from ram.**
3. **TLIST Lists all Text Files, random or sequential to screen/printer.**
4. **DUMP Memory to screen/printer in Hex with Ascii on right side.**
5. **DISA disassembles Binary to screen or printer.**
6. **AL prints last loaded program Address & Length in decimal & hex.**
7. **HIDOS moves DOS to Language Card & continues operation of program.**
8. **/ is a one keystroke Catalog in addition to the original command.**
9. **DATE prints with any clock. Also File Dating with clock or manual.**
10. **FIND prints address's of hex found in 64k memory. Hidos cmd only.**

Compatible

All DOS entry addresses have been preserved. DOS is original length and compatible with most software. David-Dos II is copyable and creates fully copyable updated disks. DAVID-DOS II is licensed by programmers for inclusion in the software they sell. Init areas were used for David-Dos II. Works with all Apple IIs including IIe 80 Col, Franklin & Hard Disks, such as Corvus & Xebec. Requires 48K. Complete documentation for screen or printing and many utilities are on the disk.

All times in seconds. (Time Test programs available)		DAVID DOS-II	ProDOS	DIVERSI DOS	DOS 3.3
TEXTFILES (100 Sectors)	TSAVE	8.0	NO	NO	NO
	TLOAD	6.2	NO	NO	NO
(791 Strings,	WRITE	29.3	28.0	29.4	88.4
32 chars ea)	READ	24.3	16.3	24.3	83.8
	PRINT/READ	44.2	45.9	45.1	117.1
(442 Sectors, 7 x 500)	APPEND	142.3	142.9	151.1	1231.2
APPLESOFT (100 Sectors)	*SAVE	7.1	16.4	6.4	33.1
	LOAD	5.0	4.0	5.0	23.5
INTEGER (100 Sectors)	*SAVE	7.3	NO	6.6	33.4
	LOAD	4.9	NO	4.9	23.4
BINARY (100 Sectors)	*BSAVE	7.8	18.4	7.3	28.7
	BLOAD	5.8	4.8	5.8	24.5
48K PROGRAM SPACE	APPLESOFT	36,352	NO	36,352	36,352
(With 3 Bufs avail)	INTEGER	36,352	NO	36,352	36,352
	BINARY	36,352	34,816	36,352	36,352
64K PROGRAM SPACE	APPLESOFT	46,592	31,232	45,658	35,162
(With 5 Bufs avail)	INTEGER	46,592	NO	35,162	35,162
	BINARY	46,592	40,704	45,658	35,162
NUMBER OF DOS COMMANDS		37	29	31	28
CLOCK FILE DATING		YES	YES	NO	NO
MANUAL FILE DATING		YES	NO	NO	NO
ONE KEYSTROKE CATALOG		YES	NO	YES	NO
AUTO USE INTEGER CARD ANY SLOT		YES	NO	NO	NO

Each Program was tested twice W/Apple Clock Card on a newly formatted disk containing DOS.
*Add 5 seconds for Verify. Apple II, Applesoft & ProDOS are trademarks of Apple Computer

To Order: Send Check or Phone Visa/MasterCard.
Add $2.00 Shipping. Overseas add US $4.00
Calif. add 6%. All orders airmailed.

$39.95

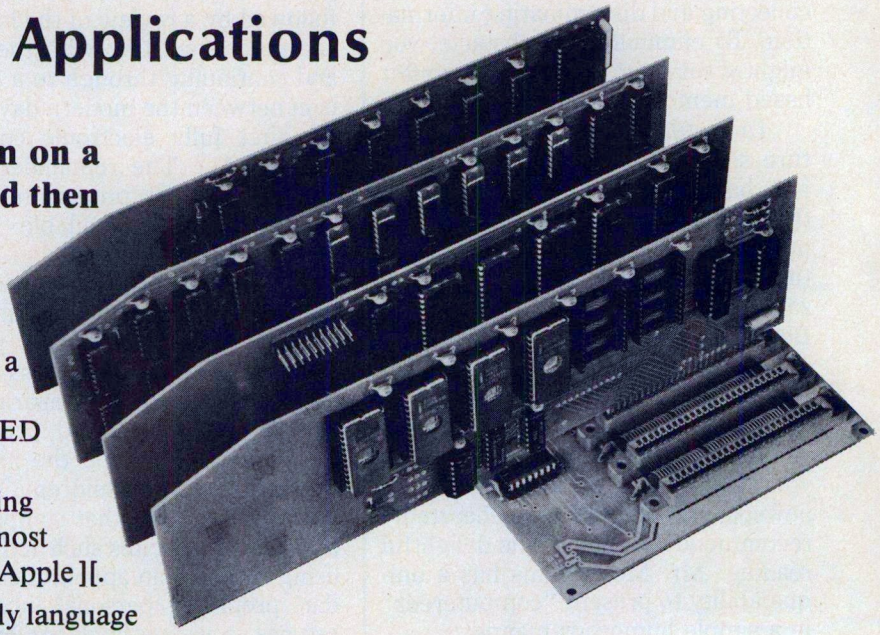
DAVID DATA™
12021 WILSHIRE BLVD., SUITE 212A
LOS ANGELES, CA 90025 (213) 478-7865

Circle 30 on Reader Service Card

APPLESEED T.M.

Apple Compatible Hardware for Dedicated Applications

Design and program a system on a standard Apple computer and then market it with the low-cost APPLESEED system . . .



Designed for use in systems where a full Apple computer may not be necessary or desired, the APPLESEED Bus system retains all the essential features of an Apple system, including mother board compatibility with almost every plug-in board available for the Apple][.

Programs are developed in assembly language on a standard Apple][. Upon completion, a set of EPROM's are blown and inserted into the APPLESEED Bus system, which then takes control of program execution.

You buy only what you need for your application; gone are the built-in keyboard, the consumer-oriented packaging and graphics subsystems. Cost, size and power consumption are substantially reduced while flexibility is greatly increased. Compatibility with a multitude of sophisticated programming and hardware add-ons, widely available for the Apple][, has been fully retained, providing a wide range of options for designing customized systems. (Applesoft PROM's are not included.)

A sampling of APPLESEED Bus components:

CPU BOARD The 6502 has been removed from the mother board and included in this card, which also contains the clock circuitry for the system.

64K RAM BOARD 64K of dynamic RAM with provision for control to be used in the normal Apple][conventions; deleting COOO-CFFF and software control over the DOOO-FFFF area.

EPROM BOARD Jumper selectable for 2716 (EPROM) or 34199 (PROM); blow 2716 assembly language programs and then run them in this EPROM board. Jumpers allow selection of software on/off, initialize on/off and other controls.

UART BOARD Software/hardware select all standard EIA baud rates. The board supports four 6850 UART's which allow four independent input and output EIA lines. Each UART is fully programmable for parity, bit length, baud rate, etc.

UNIVERSAL BOARD The APPLESEED Universal board contains the following: One 2716 socket for an F8 PROM, one 2K × 8 bit RAM (0-7FF), one UART channel for serial communications, 8 parallel INPUT lines (TTL), 8 parallel OUTPUT lines (TTL), device decode for one additional standard slot-dependent board, support for standard Apple][style keyboard, and a power supply connection. Each feature is independent and any combination may be used simultaneously. The Universal board is designed to be used with the APPLESEED CPU board.

TEXT BOARD The TEXT board is designed to add the conventional TEXT (page 1 and page 2) to an APPLESEED system. The board generates composite video to plug into a standard monitor or T.V. modulator. The board provides 40 columns × 24 rows in each page and is not slot dependent.

MOTHER BOARD 10-slot motherboard is fully compatible with most Apple add-on cards currently available.

Please call or write for prices and more detailed information on these and other APPLESEED Bus products.

DOUGLAS ELECTRONICS 718 Marina Blvd. • San Leandro, CA 94577 • (415) 483-8770

Apple][is a registered trademark of APPLE COMPUTER INC.

Circle 17 on Reader Service Card

ABC's of ABBS's

(PART 2)

philip chien

IN part 1 of the ABC's of ABBS's, I answered a number of questions about operating a bulletin board, and let you have a taste, as seen through a SYSOP's eyes, of how much time is demanded from the system operator.

Now that you've seen what it's like running a system, you may find that you want features which aren't in the commercial package you have, or you may want to write your own system from scratch, and may even want to (gasp!) try to sell it and make gobs of money. The following is a list of features I feel a BBS should contain, and features which would be nice for a BBS to have. All of them will be combined into a BBS system I'm writing, which will hopefully be available for sale later this year.

Note: I am leaving out obvious features which all (or at least almost all) bulletin boards have. When I put in a feature which should be obvious, then it is included because there are boards which don't have it. It simply would be impossible to mention all of the more common and desirable features.

Musts:

The diskettes should not be copy-protected at all, and all source code must be given to the user. This cannot be over stressed. There is no conceivable way a program diskette can survive month after month of continuous use, making continual back-ups necessary. And there is no way that the end user will be satisfied with the system as it comes "out of the box," no matter how good the original programmer thinks it is. Note that I said that the diskettes cannot be copy-protected, but the system can. Product support, including upgrades, can be protected from unauthorized copies with serial numbers, user registration, etc.

A BBS system must be able to recover from all errors. Since the system has to run by itself, you won't be able to reset from any errors. If the user is asked for a numeric input and types either a string, or a too large number (e.g. 10 E 99 — a number greater than the number of atoms in the universe), then the program must be able to recover. If the user hits a stop character, e.g. [Ctrl-S], and hangs up, then the system must also hang up and properly prepare to receive the next call. Any disk I/Os which access non-existent drives must also be recovered from. Should the program access a local printer while the system is running (e.g. print a userlog and information on the caller), then the system must recognize if the printer goes offline, or runs out of paper.

A BBS system must be able to recover from all errors.

All Bells {CHR$(7) $87} Should be trapped. Many pain-in-the-posterior callers hit the [Ctrl-G] key continuously to try to get the SYSOP's attention.

The system should not allow a new user to access private features (e.g. mail system, member lists, etc) until the user is verified by the SYSOP. Until verified, a user should only be able to send messages to the SYSOP. The SYSOP should have the capability of deciding whether an unverified user can read everything, or can only read a limited number of messages until verified.

SYSOP commands (e.g. dump userlog, access other user's mail, change users, etc.) should *only* be accessible from the local user's keyboard. If there is any remote SYSOP capability, then there should be an additional hidden SYSOP password which must be typed correctly before access is granted. Should there be an extra utility program (e.g. change passwords, verify users, etc.), it should be kept offline (not even in any drive attached to the system) whenever the program is running.

A system should allow the use of different size disk drives. The Apple Disk][just isn't cost effective beyond the first two drives, and most bulletin board applications will require more memory.

A decent system must accept, and be able to process upper and lowercase. If a person answers a Y/N question with a lowercase "y" then the BBS should accept it. Displays should be formatted so they appear in 80 columns on an 80 column screen, and 40 columns on a 40 column screen. (Who cares about users with 24, 32, and 64 column screens anyway!)

When a person calls the system, the title of the system, and possibly a short message should be displayed. The user should be asked to press return to set-up the baud rate. (In addition, many users are accustomed to pressing return when logging onto a system.) Then the user should be asked "Do you have a password on this system [Y/N]?" A "Y" or "y" response should take the user to the logon section. A "N" or "n" answer should take the user to the new user information and logon sections. If the user should type any other key, the question should be asked again. If on the second try, the user doesn't answer either "Y," "y," "N," "n," then either ask the user if he/she knows what he/she is doing, or hang up. Face it, there are people when told to answer "Yes" or "No" won't know what to do.

When logging off, the user should be asked "Are you sure?" Anything other than "Y" or "y" should send the user back to the main menu.

Should the user not type in any keystroke for two minutes, a warning message should be printed, and the user logged off if no response is

received within 30 seconds (just in case somebody falls asleep while online).

An expert mode should be available where frequent callers can eliminate the menus. The prompts should still state where in the system the user is. e.g. *Mail sub-system, Bulletin base, Discussion section, Utility section, etc.*

Never, never under any circumstances use the phrase: "Do you mind?" This is also a good rule when not on a BBS. The phrase shouldn't be ambiguous, but it is.

Each copy of your BBS which you sell should have a serial number and version number displayed when the user logs on. This helps protect you, and will help you track down problems when the SYSOP calls you, and needs help.

The system should let the user select a personal password. The password should be between 7 and 14 characters, and only *uppercase* ASCII should be allowed. If the user types in lowercase letters, they should be changed into UPPERCASE. The minimum ASCII character value accepted should be the space {CHR$(32) $A0} and the maximum should be RUB {CHR$(95) $DF}.

Should the BBS have a discussion area, the SYSOP should be able to decide, through the System Configuration menu, whether or not an end user should be able to start a new discussion, or if only he/she should be able to start a discussion.

Commands should be mnemonic, where the character relates to the command name. On one system, the letter "O" is used to logoff, whereas on another system the user types "G" for Goodbye, while still a third uses "T" for Terminate. Unfortunately there is no standard set of commands. To make it easier for a user to memorize the commands, the following rules should be obeyed:

1. Whenever possible, use the same letter for the same purpose. e.g. if you use "M" for menu at one point, don't use "Q" for quit to menu in another portion of the BBS.
2. Sometimes it is necessary to use the same letter more than once in your command set. e.g. the letter "S" is used for: Select, Scan, Skip, and Save. In these cases, the full words should be displayed any time the user might get confused about which command will be executed. Careful planning ahead of time, and deciding which character will work with which command, will prevent problems.
3. When displaying mnemonics, highlight the character used. e.g. Save should be displayed as: [S]ave, with the S in inverse. Assuming that most of your users will have a Datamedia compatible terminal (i.e. a Videx Videoterm or Ultraterm) or most CP/M terminal programs, then sending a [Ctrl-O] will put the user into INVERSE mode, and a [Ctrl-N] will change the user back to NORMAL. However, be aware that if the user doesn't have an INVERSE ROM, then all characters which you want the user to see in INVERSE will appear as white blocks. Those users should be instructed to configure their terminal programs to ignore [Ctrl-O]'s. This could also be handled through a user-accessible personal profile section.
4. While in expert mode, if the user forgets a command, then hitting the "?" key should list the valid commands.
5. As always, lowercase characters should be accepted and changed into the correct UPPERCASE commands.
6. There is no rule #6!

If these rules are followed, hopefully the user won't get confused. If the rules aren't followed, the user will be as confused as if the Shift key was placed in the wrong place on the user's keyboard.

Nice features, but not absolutely necessary:

The name of the BBS should be displayed when the user logs on. The SYSOP should also have the option of a pre-logon message.

An information file should provide a variety of information: the name of the BBS, the author of the program (and where an interested person could purchase it), the name of the SYSOP, hardware used, purpose for the BBS, and topics covered.

Most BBS's have a "chat" feature where the user can type messages back and forth with the SYSOP. In order to "call" the SYSOP, usually the bell of the local Apple is rung. Unfortunately this has a tendency to wake sleeping SYSOPS. An In/Out flag which the local user can set, eliminates this problem. If a user tries to chat while the SYSOP is out, then a message like "I'm sorry XYZZY is out right now. Please try chat later" should appear on the users terminal, and the bell should *not* be rung. As a bonus, in the system I'm writing, an LED attached to the Game I/O will flash when the user tries to chat, whether or not the SYSOP is actually in or out. This way the SYSOP can decide whether or not to talk to the user. The LED is also quiet, so it doesn't disturb anybody, but is sufficient to attract the SYSOP's attention.

When a user calls the system, the SYSOP's display should include the user's name, security level, etc. This information should remain on the screen outside the TEXT window until the next user calls. (That way, if a user "crashes" the system, you can find out who did it.)

Clock cards should be supported, but the system should work without a clock installed. When files are timestamped, they should not say "DATE MAILED: NO CLOCK AVAILABLE" Rather, they should ignore the timestamp routine. If a user has a clock, then while the system is waiting for a call, the time should be printed on the screen. If you want to get really exotic, the system should also be usable as an alarm clock, and set off a BSR-X10 when requested.

Glossary of BBS terms and commands:

Abort: Stop what you are doing. Return to the menu you came from.

access level: Many systems have separate sections of the board that can only be accessed if you are a "member" of the group. e.g. a dating system can have an open section, and adult section, and a couples section. If these sections are arranged as tiers where the higher tiers can access each of the lower tiers (e.g. an adult user also has access to the open section, and a couples user has access to all three), then the sections are referred to as access levels. The higher the access level, the greater the number of features available.

answerback: In the mail system, after you read a letter and wish to respond to the same person, answerback will automatically send the next letter you write to the person who wrote the letter you just read.

auto-dialer: Something the end user should be concerned about. With a popular board, it may take several phone calls to get through. An auto-dialer will keep dialing until it gets a carrier, and will then enter the terminal mode. Several popular communications packages (i.e. ASCII Express) have auto-dialers built in.

BBS: Bulletin Board System – (read this article)

Carrier: One who carries. Also the high pitched tone (it does not warble, contrary to what was shown in the film "Wargames") which a computer sounds to indicate that it is ready to be connected to the user.

Chat: The function which lets the user communicate with the SYSOP. The users communicate by typing back and forth, not talking. The best protocol used is: first the SYSOP types a "Hi there" message and at least two carriage returns, then the user responds and back and forth until there is nothing left to say. Usually only the SYSOP can end the conversation and return the user to normal bulletin board operations. In more cases than not, it's easier for both users to pick up the phone, and shut off the computers, but it's fun to type messages back and forth.

Clear screen character: This character will clear the user's screen. In the majority of cases it will be a Form Feed {CHR$(12) $8C}. If a printer is online, then the printer will be advanced to the start of the next page.

co-sysop: (see sub-sysop) **

Cole's Law: thinly sliced cabbage

default: Fortunately not what it means in the business world. The default response is what the user gets if he/she hits the return key. e.g. Are you sure Y/N (default Y) if the user hits return, then the computer assumes that the user is sure and continues. The default response in this example is "Y".

DFT X-MODEM CRC-16: These are various popular "protocols" used for transferring data. Normally only text characters can be transmitted. This is good enough to download text, but to upload text, or to transmit programs, normally a protocol is needed. To receive a program via protocol, a terminal program with the same protocol must be used. There is one unique download protocol for the Apple without a terminal program. Should the user have a Apple Communications card compatible modem (e.g. Micromodem, Apple Cat with program ROM, etc.) then the system can automatically download a program to the user's machine, and save it on disk.

down: Opposite of up. Also, (As in "The system is down now") The system is unavailable, and you can't call it. Many BBS's are down more than up.

download: The process where a user obtains a program from the BBS system for eventual use on his/her own system.

Editor: The person who decides whether or not you are reading this article. Also, when online, the subroutine that lets you correct mistakes. Don't expect much of an online editor though, most online editors are limited to a line editor.

E-mail: Electronic Mail. A system of sending private text files back and forth to specific users.

feature: (As in "Call my BBS and see all of the *features* that it has.") A feature is a bug that has been documented. (e.g. a feature of some BBS's is if you type the word "OFF" anywhere, you will be logged off.)

FILO: First In Last Out. The first message put in will be the last message read. When you read messages, they will be read in reverse order. That way you always read the most recent messages.

kill: Fortunately not what it means outside of the computer industry. To remove. On some systems a user can remove his/her own public messages

The $175 disk drive for Apple II

Now, you can have a full-sized 5¼" floppy disk drive compatible with the Apple II for the unbelievably low price of $175.
Also, check out these half-height 5¼" floppies:

- Single Apple half-height $195.00
- Dual Apple half-height $390.00

(includes the drive, cabinet, cable, and our standard warranty including 90 days parts and labor)

AVAILABLE NOW FOR DELIVERY

TELEPHONE ORDERS ONLY
MASTERCARD, VISA, CASHIER'S CHECKS COD ORDERS ACCEPTED
(818) 883-0587

StarLogic

20860 Plummer Street
Chatsworth, CA 91311

Apple is a registered trademark of Apple Computer, Inc.

(Also compatible with Franklin ACE)

Circle 12 on Reader Service Card

to other users. On others only the SYSOP can kill a message. Normally it is fairly difficult to "un-kill" a message.

last call: (also last on) This date is the last time that you accessed this particular system.

last user: The last person to access this BBS.

logon: The sequence which is used to enter the BBS.

logoff: The procedure to hang up properly.

Mr. Sunshine: A term used to describe a user disliked by other users, usually a bore, or a person who tries to pick fights.

nulls: These are "empty" characters which are used to slow down the display. If you have a printer printing everything as it comes down from the BBS, and the printer isn't fast enough, or doesn't have enough of a buffer then you'll lose the first few characters of each line. If nulls are sent then the characters won't be lost. At 1200 baud, almost all printers will require nulls to avoid losing characters. The null character has an ASCII value of zero.

post: To place a message where other users can read it.

BYTES & PIECES **PRICE RIOT** Ask For Free Price List

$1.00 credit for phone orders over $100.00

	LIST	SPECIAL
Bookends	125.00	90.46
Computer Baseball	39.95	28.56
Crossword Magic	49.95	35.96
D.B. Master 4.0	350.00	252.96
Ess. Data Dup. (EDD)	79.95	58.16
Flight Simulator II	49.95	36.96
Fontrix	75.00	54.46
G.P.L.E.	49.95	35.46
Graphics Magician	59.95	43.36
Home Accountant	74.95	53.46
Homeword	69.95	49.96
Incredible Jack	179.00	130.46
Koala Pad	124.95	90.46
Krell Logo	89.96	75.46
Krell SAT	299.95	252.96
New Step by Step	89.95	64.56
PFS File (II + or IIe)	125.00	90.46
PFS Write	125.00	90.46
Planetfall	49.95	35.46
Sensible Speller	125.00	90.46
16 K Card (Microsoft)	99.95	75.96
Think Tank (II + or IIe)	150.00	108.96
Trains	39.95	28.56
Wildcard II	139.95	110.19
Wildcard +	169.95	133.56

Ask for our educational software catalog

C.O.D. • M.O. • Cert. Checks • School P.O. • VISA • MasterCard • Checks Allow 2 Weeks • N.Y.S. Res. Add Sales Tax • US Orders Under $150. Add $2.00 P & H • All Canadian, US Funds $3.00 P & H. • Foreign, Charges Only, Min. P & H $7.00

Bytes & Pieces, Box 525, Dept. C, East Setauket, N.Y. 11733, (516) 751-2535

Circle 13 on Reader Service Card

rb: Ring Back. In cases where the SYSOP cannot afford to call, or put in a second phone line, a ring back is often used. Should a user call and let the phone ring, then a normal (voice) answering message will take the call. All users of the bulletin board are instructed to use the following procedure:

1) Call the phone number and hang up after one ring.
2) Call again within thirty seconds.

The voice phone answering machine is set to answer on the third ring, the BBS is set in ring back mode. (Naturally the BBS software and hardware must be able to support ringback). This way, normal phone messages can be received from non-computer owners (read: people still left in the stone and transistor age) as well as people who want to access the bulletin board.

scan: Grammatically incorrect, it should be skim. Scan commands usually just displays the titles of the messages. This way the user can select which messages are interesting without wasting time reading the other messages.

so (usually lowercase): sexually oriented. The board has sex topics. This can mean either sleazy X-rated dime novels, or interesting adult sexually oriented, as well as other discussions.

sub-sysop: (see co-sysop)

syscrash: When something doesn't work right, and Murphy strikes. Translation: that cute user to whom you wrote a letter asking for a date didn't get your letter.

SYSOP: SYStem OPerator — the person who runs the system. Also referred to as the great one, the most magnificent person alive, Ghod, etc.

timestamp: The subroutine which checks the time, and puts the time into the message sent by the user.

toggle: When the same keystroke that turns a command on is used to shut it off, it is referred to as a toggle.

userlog: A list of the most recent users to access the system. Usually it is in FILO order.

video width: This command, which isn't available on all systems, and on the systems where it does exist, normally affects only some of the displays, lets you set the maximum video width to the video width of your screen. (On an Apple normally 80, unless you don't have an 80 column card).

Wizard: (see under Philip Chien)

xpert mode: Spelling for Expert mode. When the expert mode is on, time-consuming menus are not displayed. After using the same system several times, you'll remember the commands you normally use. The Xpert mode when selected will save you a lot of time.

Y/N: When used at the end of a question, it indicates that the system is asking a Yes-No question, and requires a response of "Y" or "N".

24 hour system: The system is in operation *most* of the time, not 24 hours a day. When the SYSOP isn't using his/her computer, the bulletin board is running.

300 baud: This system uses the Bell 103 protocol at 300 baud. Baud is a speed unit equivalent to a bit per second. 300 baud is 300 bits per second, or 30 ten bit characters per second. The ten bits are usually eight data bits and two stop bits, but can also be other permutations of start, data, and stop bits.

300/1200: This system will operate at both 300 and 1200 baud, using the Bell 103 and Bell 212 protocols. When you connect, the system will automatically determine what rate you're calling at and set itself at the same rate.

Seriously, co-sysop and sub-sysop are usually used interchangably. But normally a Co-sysop has as much control of the system as the SYSOP has, where a Sub-sysop takes care of and maintains the files in one section of the system (this can be as much as posting material, editing other users material, and controlling what goes on, to as little as just a figurehead title).

Questions seen often on BBS's

Are you sure? This doesn't mean what it says. Some of us are never sure. "Are you sure" means do you want me to continue with what you just asked me to do.

Shall I use your last message? This feature, means use the same message that you typed in last. It's very convenient for sending the same message (i.e. a form letter or generic Christmas card) to many users.

LOGON: The proper answer to this question is "JOSHUA." Whether or not it does anything is a different question. It may play "Global Thermonuclear War," or do nothing at all.

FROM THE VERY CORE OF APPLE:

DMA Protocol

peter baum
apple computer inc

THIS paper explains the hardware protocol for doing direct memory access (DMA) on the Apple *IIe* and Apple][. It is meant to serve as a guideline for developing peripherals which do DMA on the Apple *IIe* and Apple][and not as a specification for future Apple products or revisions. The reader should be very familiar with either the Apple][+ or the Apple *IIe*, especially the timing on the data and address busses in relation to the 6502.

DMA is used by peripheral cards in the Apple][to transfer data directly into memory without benefit of the processor. Transfer of data from a peripheral device into RAM can normally be handled one byte at a time under control of the processor. By using DMA, data transfer rates can be achieved which are greater than the 6502 can handle in software. This transfer rate can approach the full cycle time of the memory. This technique can also be used to transfer single data bytes into memory without requiring the CPU to process an interrupt, which can be very time consuming.

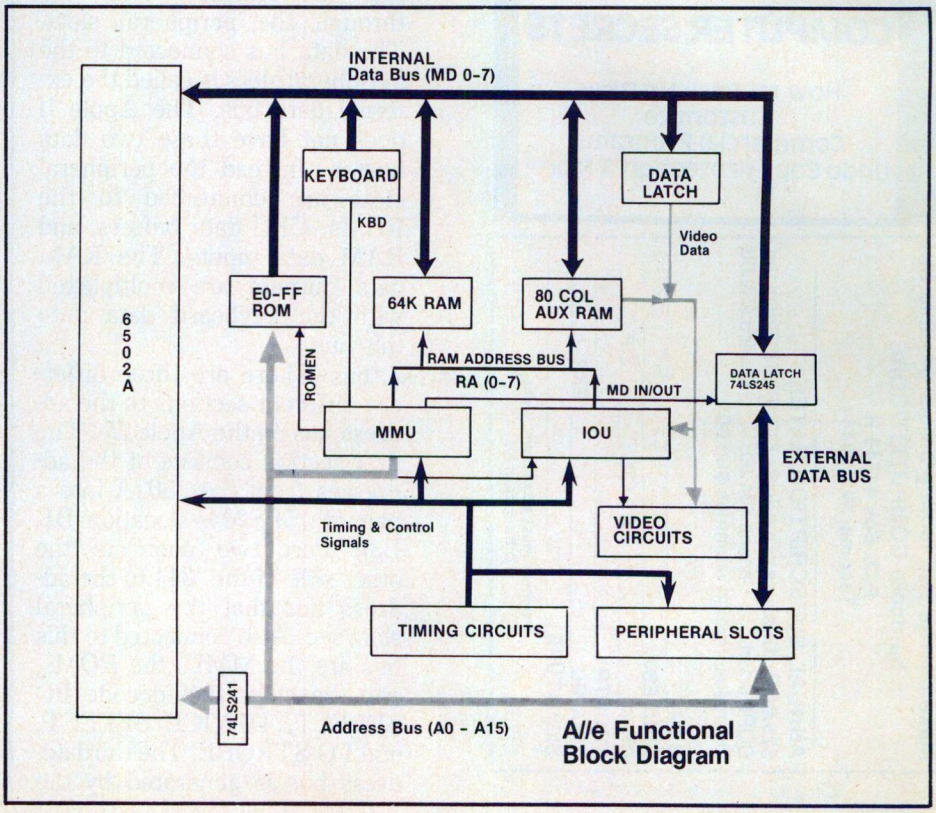
The DMA process entails five steps: turn the processor off, gain access to the R/W* line and both address and data busses, complete the data transfer, release the data and address busses, and finally, allow the microprocessor to start back up. Each of these steps is covered in detail in the following paragraphs.

At this point, I should caution the prospective designer that DMA on an Apple][+ or Apple *IIe* can only be done under certain circumstances. Because DMA turns off the processor, any program with a software timing loop will not work properly. These programs assume that each instruction will take a fixed amount of time, which is not true when the processor stops in the middle of an instruction. This means that the Apple][disk drives will not work since they require a timing loop to read a disk. (Co-processor cards work with DMA because they initiate the disk access and know that DMA can't be used until the disk is finished).

Another problem is that because of the mapping scheme used on the Apple *IIe* extended 80-column (64K) card, a peripheral card can not tell which memory bank is being used without a complicated detection scheme. This means that if a DMA device wrote to a certain memory space, it might not be able to read the same data back.

Though the differences between the Apple *IIe* and Apple][+ architecture appear to be significant to a device which uses DMA, this should not affect the design in most cases. A good rule of thumb is that if a device is designed to work on the Apple *IIe*, then it will be backward compatible and also run on the Apple][+. The converse is not true, cards that use DMA on the Apple][+ might not work on the Apple *IIe*. Hence, most of the descriptions in this document refer to the Apple *IIe* with occasional references to the Apple][+. For example, the timing specifications listed are calculated from the Apple *IIe* timing paths unless otherwise noted.

Occasionally the descriptions refer to a chip on the motherboard of the Apple *IIe*, so a set of Apple *IIe* schematics should be handy. The corresponding parts on the Apple][+ will be specified where applicable.



A/Ie Functional Block Diagram

We Are Not
PIRATES
We simply prefer software that is
NOT LOCKED UP!
If you do, too, then subscribe to:
core
IN-DEPTH COVERAGE
of
Special Interest Subjects
• GRAPHICS • DATA BASES
• UTILITIES • GAMES
For Hobbyists and Advanced Computists
hardcore/CORE
2-in-1 published monthly
hardcore
HARD FACTS
on
APPLE SOFTWARE
An Unprecedented and Controversial disclosure of closely-held
COMPUTER SECRETS
How to: Back-Up Disks
Customize
Commercial Programs
Undo Copy-Protection & More!
Make checks and money orders payable to:
hardcore COMPUTIST/CORE
P.O. Box 44549 H
Tacoma, WA 98444
ANNUAL SUBSCRIPTION RATE:
$25.00 for 12 issues of hardcore COMPUTIST with the CORE insert. Canada:$34.00 Mexico:$39.00
Overseas, foreign countries $60.00
NAME
ADDRESS
CITY STATE ZIP
COUNTRY EXP
VISA/MC
SIGNATURE
Payment must accompany order.
U.S.funds only.
Circle 14 on Reader Service Card

The following paragraphs describe and define some of the terms that are used throughout the paper. The Apple *IIe* block diagram on the next page may be helpful when reading about the busses.

01 (phase one) time–The time when the 01 system timing signal is high. During this time the data bus, address bus, and RAMs are used for video refreshing. This time is also called the video cycle or video phase.

0o (phase zero) time–The time when 0o clock is high. 0o is the inverse of 01. During this time the microprocessor uses the data and address busses. This time is also known as the CPU cycle or CPU phase.

IOU and MMU–The MOS custom chips inside the Apple *IIe*. See chapter 7 of Apple *IIe* Reference Manual for more details on the custom chips.

data bus–The microprocessor, ROMs, and RAMs are connected to this bus. On the *IIe* this bus generally has MOS components connected to it rather than TTL and is sometimes called the MOS data bus. A 74LS245 bi-directional bus transceiver (location B2 on motherboard) connects this internal bus to an external bus that the outside world sees through the peripheral slots. The data bus connected to the peripheral slots is called the external data bus. The Apple][does not have these two data busses. Instead the peripheral slots are connected to the ROMs, CPU data buffers, and RAM data inputs. The RAM data outputs are multiplexed with the keyboard data onto this bus.

address bus–There are three different different sections to the address bus on the Apple *IIe*. The first section consists of the addresses from the 6502A into a pair of 75LS244s (location B1, B3). Part two connects the other side of the '244 to the address bus that the peripheral slots see. Also connected to this bus are the MMU, the ROMs, and the chips that decode I/O SELECT, DEVICE SELECT, and I/O STROBE. The third address bus is generated by the custom chips and is only used to access the RAMs. the MMU and IOU automatically multiplex this bus with the high byte and low byte of an address during any RAM access, whether it be for video refresh or for a microprocessor instruction fetch. This third bus is called the RAM address bus. The Apple][also has these three busses but uses 8T97s and discrete logic instead of the 74LS244 and custom chips.

6502 microprocessor–In the Apple *IIe* a 6502A, a 2 MHZ. part is used instead of the 1 MHZ. 6502 used in the Apple][+. Since the custom chips in the Apple *IIe* are MOS and slower than the TTL in the Apple][+, the faster 6502A was used to guarantee better margins. For example, the 6502A sets up the address bus faster on the Apple *IIe* than the 6502 does in the Apple][+.

Getting on the Bus (Exact Change Only!)

1) Pull DMA low during 01 time

On the Apple *IIe*, the DMA line controls the direction of the 74LS245, which enables the internal data bus outwards to the peripheral slots or enables external data onto the internal bus. Changing the state of the DMA line during 0o could cause the '245 to change directions, forcing the internal data bus to go tri-state during a microprocessor read. The 6502 would read garbage and the computer might go belly up by jumping to a random memory location.

On the Apple][, pulling the DMA line always forces the CPU data bus buffer to point inward and drive toward the 6502. Pulling the DMA line low during 0o of a write cycle would result in garbage being written to memory, since the data bus to the RAMs would suddenly go tri-state.

2) Wait 30 ns., then assert address bus and R/W* line

Before driving the address bus and R/W* line the system must process the transition on the DMA line and release the bus. This requires:

25 ns. 'LS244 output disable from low level

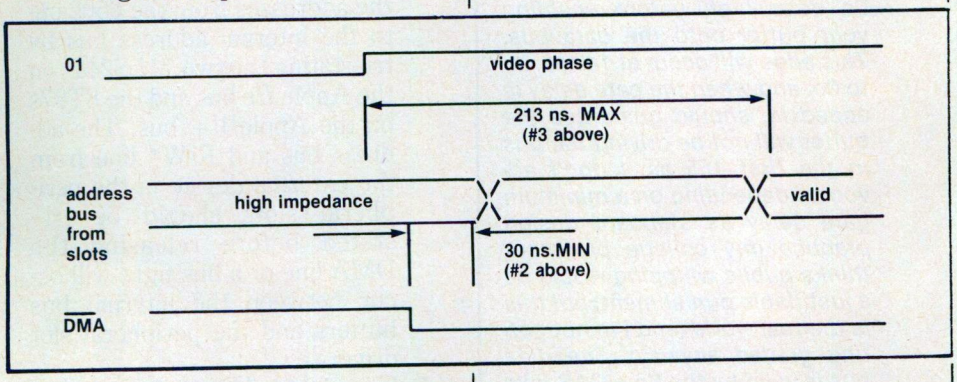
5.ns. S02 low to high level output transition

30 ns. delay from DMA negative edge before driving address bus

The 30 ns. wait will also work on the Apple][, since it only needs 27 ns. ('LS04 and 8T97).

3) Address and R/W* line must be valid within 213 ns. of 01 positive edge

This constraint is needed to meet the set up requirements of the IOU, MMU, and RAMs. This time can be derived from the 6502A (2 MHZ) set up requirements. The Apple][can wait for 300 ns. before data must be valid, because it uses the 1 MHZ. 6502 which has a longer set up time.



During DMA (Keep your hands inside the bus at all times)

1) Don't drive data bus during 01 time!

On an Apple][+, it is safe to drive the data bus 35 ns. after asserting the R/W* line low, regardless of the point in the timing cycle. When the R/W* line goes low, the 74LS257s at locations B6 and B7 tri-state the data bus, even in the middle of 0o and 01. This prevents a bus fight from occurring between a DMA device and the system.

At a first glance of the Apple *IIe* logic schematics, it appears that a bus fight cannot occur on the data bus. During the 01 half of a write cycle, the 74LS245 tri-states the data bus within 30 ns. of the R/W* line pulled low. While this does preclude a fight from occurring on the data bus during 01, it doesn't prevent a bus crash from occurring at the beginning of 0o. At the beginning of 0o, the 74LS245 is enabled and will drive the external data bus. If the peripheral card also drives the data bus, there could be a horrendous bus fight since the 74LS245 can source 15 ma. and sink 24 ma. per line. This might cause a spike on the ground plane, which could cause a processor to reset on a co-processor card.

Let us take a look at the problem by stepping through the following timing diagram:

#- Apple][+ specification

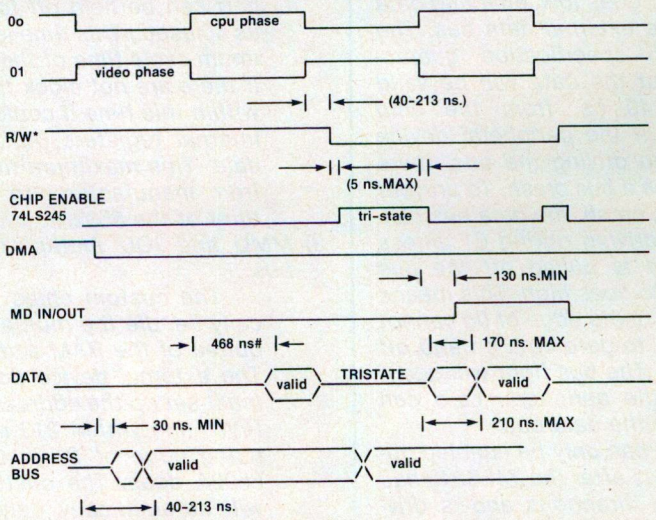
01	R / W *	stable state of 74LS245
		10 (write toward RAMs) high impedance
		11 (read from RAMs) outward driving external data bus (slots)
		00 (write toward RAMs) inward driving into RAMs
		01 (read from RAMs) outward driving external data bus (slots)

Table 1 *Stable State of 74LS245*

The diagram starts with the video cycle of a read operation. During the video cycle, the video refresh data is read from the RAMs and put on the data bus. This video data will appear on the peripheral slot (external) data bus because the 74LS245, as can be seen from table 1, drives outward during 01 of a read cycle.

Typically, the address bus and R/W line would be set up by the 6502A during 01 for the next CPU cycle, but instead a peripheral card pulls the DMA line low. As explained earlier the peripheral device should wait at least 30 ns. before driving the address bus and R/W* line. In this first DMA cycle the peripheral card wants to read a byte from RAM, so it keeps the R/W* line high.*

At this point I must switch over and use the Apple][+ to explain the timing required to read the data from RAM. The rule of thumb, that designing a DMA card for the Apple //e will be backward compatible and run on the Apple][+, will not hold here. On the Apple][+ data is valid on the peripheral connector a minimum of 468 ns. from the 0o rising edge and holds to at least the falling edge of 0o at 490 ns. The hold time is actually the minimum propagation delay from the falling edge of 0o through the following chips: 74LS257 at J1, 74LS139 at F2, 74LS20 at D2, 74LS00 at A2, and finally to the



#- Apple][+ specification

enable of the 74LS257s at B5 and B6. On the Apple //e a byte from RAM becomes valid at most 345 ns. after the rising edge of 0o and stays valid until the 0o falling edge.

In the second DMA cycle, the timing diagram shows the peripheral card writing a byte to memory. In the first phase of the cycle, the video phase, the address bus and R/W line are set up by the peripheral card within the timing specifications described earlier, 213 ns. Though the direction of the 74LS245 still points toward the slots, the '245 is disabled when the R/W* line is pulled low by the peripheral device (see table 1). This will tri-state the external data bus. All the signals stay unchanged through the rest of the video phase, until the CPU cycle starts with the rising edge of 0o.*

Most bus fights occur at the beginning of the CPU cycle. The CPU cycle begins with address bus and R/W line set up already and the data bus tri-stated. The signal MD IN/OUT, which drive the 74LS245 direction control, is generated by the MMU and is always low during 01, so that the 74LS245 drives toward the slots. MD IN/OUT uses the 0o rising edge to clock itself high during a DMA write cycle, but because the MMU is an MOS chip the delay before MD IN/OUT finally rises can be as long as 130 ns. from the 0o rising edge. Hence, at the beginning of 0o the 74LS245 is in tri-state mode but with the direction set to drive toward the peripheral slots.*

Within 5 ns. after 0o goes high, the chip enable to the 74LS245 goes low, enabling data onto the external data bus. The 74LS245 specification guarantees that the data will be valid within 40 ns. from the chip enable. If the peripheral device was also driving the bus, there would be a bus crash. To prevent this bus crash, the data bus cannot be driving during 01 unless the data is pulled off the bus before 0o goes high. This means that the rising edge of 0o cannot be used to gate data on and off the bus. The bus fight will occur before the peripheral card can tri-state the data bus.

Data can only be enabled onto the bus after the 74LS245 has changed directions and is driving the internal data bus. The DMA device must allow 130 ns. for the MD IN/OUT line to change, plus the delay for the 74LS245 to change directions which takes 25 ns., for a total of 155 ns.

After this 155 ns., the data must be valid on the bus within 55 ns. because the RAMs require data be set up at the CAS falling edge, which occurs 210 ns. into 0o. This does not leave any time to spare since, for example, a 74LS245 has a 40 ns. enable time. This timing criteria will also work for the Apple][+ since the set-up time for 16K RAMs is the same as the 64K RAMs, and CAS also falls at 210 ns. The data hold time of 55 ns. after CAS falls is also the same for both the Apple //e and the Apple][+.

Here is a scenario for a DMA write. Set-up the address bus and R/W line within the required 213 ns., then wait for the first 7M (pin 36 on slot) falling edge after 0o goes high before enabling your buffer onto the data bus. This edge will occur at 140 ns. into 0o, and when the gate delay is added in, should guarantee the buffer will not be driving the bus in the first 155 ns. I don't advocate depending on a minimum gate delay as standard design practice (my college professor thinks public whipping would be a justifiable punishment) but this is the real world (and I'm not getting graded anyway). The data bus is valid by the time CAS falls and should be stable for at least another 55 ns. or until 0o falls.*

2) The processor can be held off for a total of 10 us. (10 0o clock cycles)

This is true if a Rockwell 6502 is being used. (A Synertek part can be held off for as long as 40 usec.) This time is the maximum cycle time of the 6502 and if there are not clock transitions within this time it could result in internal registers (A,X,Y) losing data. This maximum time varies from manufacturer to manufacturer of the 6502.

3) MMU and IOU multiplex address bus

The custom chips automatically handle the multiplexing required of the RAM address bus. The external device doing DMA must set up the address bus and R/W line within 213 ns. of the rising edge of 01 just like the 6502A does. The custom chips will automatically generate the addresses to the RAMs for the video refresh cycle during 01, but use the addresses from the address bus to set up for the next instruction cycle. Hence, the only consideration on the address bus during DMA is to meet the 213 ns. set up time requirement.*

The 213 ns. set up time will also work with the Apple][since it can take as long as 300 ns. to set up the address bus and R/W line.*

Getting off the Bus

1) Don't release DMA during 0o.

This is analogous to step 1 of Getting on the Bus. If the DMA line is released during 0o, the microprocessor will try to execute a cycle during this time without the data or address bus set up properly. This random instruction fetch will probably cause the system to crash.

2) Tri-state address bus drivers on peripheral slots.

The DMA line is holding off the addresses from the 6502 onto the internal address bus by tri-stating the two 74LS244s on the Apple *II*e bus and the 8T97s on the Apple][+ bus. The address bus and R/W* line from the external device in the peripheral slots should be tri-stated before releasing the DMA line or a bus fight will occur between the internal bus buffers and the peripheral slot drivers.

3) Release DMA Line

These last two steps are the opposite of the first two steps required to get on the bus. Both of these steps, releasing the address and R/W* lines then the DMA line, should be done within 178 ns. of 01 going high. This allows time for the 6502A to set up the address and R/W* lines properly for the next instruction cycle.

213 ns.	address set up requirements
– 5 ns.	:S02 output hi to low transition
– 30 ns.	'LS244 out enable time
178 ns.	to release DMA line and allow 6502 to set up address bus

Again, the Apple][can take longer, up to 260 ns., before releasing the DMA line.

A Call –A.P.P.L.E. TECHNIQUE:

Setting I/O Hooks in ProDOS

cecil fretwell

AS is the case with DOS 3.3, ProDOS lets you change the I/O hooks to send characters to a specified device or receive characters from a specified device. How to do this under ProDOS is explained in *Apple's Technical Note #4,* 26 July 1983. Unfortunately, that note is not complete as this short paper will show.

It provides a good basis for this discussion, and was published as follows:

> "When programming under DOS 3.3, if you wished to change the I/O hooks, all that was necessary was to install your I/O routine addresses in the character-out vector ($36–$37) and/or key-in vector ($38–$39) and notify DOS (JSR $3EA) to take your addresses and swap its intercept routine addresses in.
>
> "Under ProDOS, there is no instruction installed at $3EA at all. So what do you do?
>
> "Just leave the ProDOS Basic Command Interpreter's intercept addresses installed in $36–$39 and install your I/O addresses in the global page at $BE30–$BE33. $BE30–$BE31 should contain the output address (normally $FDF0, the monitor COUT1 routine), and $BE32–$BE33 should contain the input address (normally $FD1B, the monitor KEYIN routine).
>
> "By keeping these vectors in a global page, a special routine for moving the vectors is no longer needed, thus, no $3EA instruction. Just install the addresses at their destination yourself."

As a simple example to attempt to apply the concepts of this technical note, suppose you want to trap a keyboard character and do some special processing before you pass it to the regular key-in logic. Let's suppose your special processing starts at the label SDUMP. Your install logic may simply be:

```
LDA  #<SDUMP
STA  $BE32
LDA  #>SDUMP
STA  $BE33
RTS
```

as dictated by the APPLE Technical Note. In immediate mode or using CHR$(4) in a program, you perform a BRUN to install the logic. It doesn't work. On the other hand, if you BLOAD the code, CALL -151 to switch to the monitor, then execute the code with the G command (you could just CALL the code), it works.

Now change the install code to read as follows:

```
LDA  #<SDUMP
STA  $38
LDA  #>SDUMP
STA  $39
RTS
```

In immediate mode or using CHR$(4) in a program, you perform a BRUN to install the logic. It works. On the other hand, if you BLOAD the code, CALL -151 to switch to the monitor, then execute the code with the G command (you could just CALL the code), it doesn't work.

In the first example, ProDOS BASIC.SYSTEM copies $BE30–$BE33 to $36–$39, executes the command, then copies $36–$39 to $BE30––$BE33. Your install code has been completely ignored.

In the second example, the same thing is done, except that after the final copy back to $BE30–$BE33, the Applesoft execution of the CALL statement finally alters $BE30–$BE33.

It is only a matter of Mutatis Mutandis. To make something work, it is merely changing what needs to be changed.

ACCESS ALL MICRO
11306 Southland Road
Forest Park, Ohio 45240

TOTAL ACCESS TO
Software
- BUSINESS
- EDUCATIONAL
- GRAPHICS/ UTILITIES
- PERSONAL PLANNING
- WORD PROCESSING
- BOOKS
- ENTERTAINMENT

Hardware and Accessories
- PRINTERS
- DISK DRIVES
- INTERFACES
- MONITORS
- MODEMS
- DISKS
- RIBBONS

CALL
1-800-543-1114
Ohio · Alaska · Hawaii
Call 513-825-5803
CINCINNATI CUSTOMERS USE OUR SOFTWARE PICK-UP FACILITY

Circle 15 on Reader Service Card

Apple Mechanic's hi-res type routines and fonts are usable in your programs WITHOUT LICENSING FEE. Just give Beagle Bros credit on your disk and documentation.

APPLE MECHANIC

HI-RES SHAPE EDITOR / TYPE FONT DISK
by BERT KERSEY

$29.50: Includes Peeks/Pokes Chart & Tip Book #5.

SHAPE EDITOR: Keyboard-draw hi-res shapes for animation in your Applesoft programs. Access & create **proportionally-spaced hi-res Typefaces** with each character re-definable as you want. Six fonts are included on the disk. Excellent LISTable Applesoft demos show you how to animate graphics and create professional-looking Charts and Graphs.

BYTE-ZAP: Rewrite any byte on a disk for repair or alteration. Load entire sectors on the screen for inspection. **Hex/Dec/Ascii displays** and input. Educational experiments included for making trick file names, restoring deleted files, changing DOS, etc.

MORE: Useful music, text and hi-res tricks for your programs. Clear educational documentation.

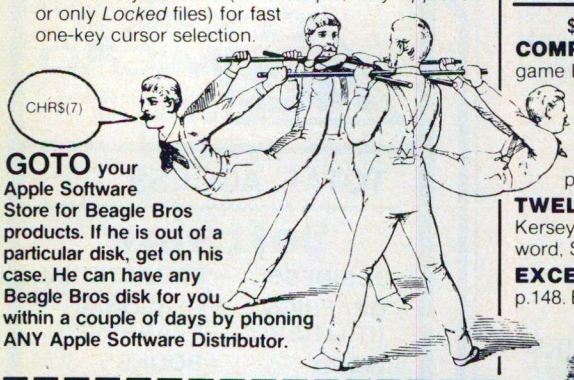
APPLE MECHANIC TYPEFACES

by BERT KERSEY

$20.00: Includes Peeks & Pokes Chart.

26 NEW FONTS for use with Apple Mechanic programs. Many different sizes and typestyles, both ordinary and Artistic. Every character—from A to Z to "★" to "□"—of every typeface—from "Ace" to "Zooloo"—is re-definable to suit your needs. All typefaces are **proportionally spaced** for a more professional appearance. People *do* notice the difference!

BEAGLE MENU: Display only the file names you want from your disks (for example, only *Applesoft* or only *Locked* files) for fast one-key cursor selection.



GOTO your Apple Software Store for Beagle Bros products. If he is out of a particular disk, get on his case. He can have any Beagle Bros disk for you within a couple of days by phoning ANY Apple Software Distributor.

RUSH the following disks by First Class Mail—

☐ Alpha Plot	$39.50	☐ Frame-Up	$29.50
☐ Apple Mechanic	29.50	☐ GPLE	49.95
☐ A.M.Typefaces	20.00	☐ ProntoDOS	29.50
☐ Beagle Bag	29.50	☐ Silicon Salad	24.95
☐ Beagle BASIC	34.95	☐ Tip Disk #1	20.00
☐ DiskQuik	29.50	☐ Utility City	29.50
☐ DOS Boss	24.00	☐	
☐ Double-Take	34.95	☐ ADD ME to mailing list.	
☐ Flex Type	29.50	☐ ALREADY ON mail list.	

AT YOUR APPLE DEALER NOW!
Or order directly from Beagle Bros—

VISA — master charge

Visa/MasterCard or COD, call **TOLL-FREE**

Orders only / ALL 50 STATES / 24 Hours a Day

1-800-227-3800 ext. 1607

OR mail U.S Check, Money-Order or Visa/MC# to **BEAGLE BROS**, 8th Floor, 4315 SIERRA VISTA, SAN DIEGO, CA 92103

Add $1.50 First Class Shipping. Any-Size Order. Overseas add $4.00. COD add $3.00. California add 6%.
ALL ORDERS SHIPPED IMMEDIATELY.

AD#8A

DOS BOSS

DISK COMMAND EDITOR
by BERT KERSEY and JACK CASSIDY

$24.00: Includes Peeks/Pokes Chart & Tip Book #2.

RENAME DOS COMMANDS & Error Messages—"Catalog" can be "Cat"; "Syntax Error" can be "Oops" or almost *anything* you want it to be.

PROTECT YOUR PROGRAMS. An unauthorized Save-attempt can produce a "Not Copyable" message, or *any* message you want. Also easy List-Prevention and other useful Apple tips and tricks. Plus one-key program-execution from catalog.

CUSTOMIZE DOS. Change the catalog Disk Volume heading to your message or title. Omit or alter catalog file codes. Fascinating documentation, tips and educational Apple experiments.

ANYONE USING YOUR DISKS (booted or not) will be using DOS the way YOU designed it.

```
10 LIST: LIST: LIST: FOR ZZ=PEEK(175)+PEEK
   (176)*256+36 TO 3072: POKE ZZ,216: NEXT
20 FOR XXX=1 TO 2: POKE-16299,0: POKE
   -16300,0: XXX=1: NEXT: REM Experiment
   with different length variable names.
```

BEAGLE BAG

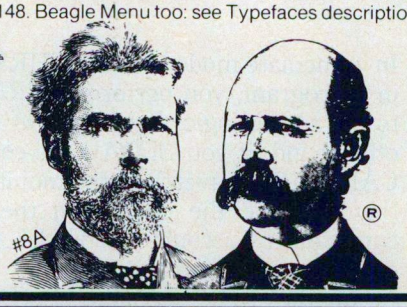
12 APPLE GAMES ON ONE DISK
by BERT KERSEY

$29.50: Includes Peeks & Pokes Chart

COMPARE BEAGLE BAG with any single-game Locked-Up disk on the market today. All 12 games are a blast, the price is a bargain, the instructions are crystal clear, *and* the disk is COPYABLE. You can even *change* the programs or list them to learn programming tricks by seeing how they work.

TWELVE GAMES from the Applesoft Ace, Bert Kersey— TextTrain, Wowzo, Magic Pack, Buzzword, Slippery Digits, and many many more...

EXCELLENT REVIEWS—See Jan-83 *Softalk*, p.148. Beagle Menu too: see Typefaces description.



Beagle Bros
Micro Software Inc.

4315 SIERRA VISTA / SAN DIEGO, CA 92103
619-296-6400

ALL BEAGLE DISKS ARE UNLOCKED, COPYABLE AND COMPATIBLE WITH APPLE II, II+ AND IIe.*
(Don't Settle for Less!)

*DISKQUIK requires Apple IIe.

"APPLE" is a Registered Trade Mark of You-Know-Who.

NEW! SILICON SALAD

INCLUDING TIP DISK #2
by BERT KERSEY and MARK SIMONSEN

$24.95: Includes Peeks/Pokes AND Commands Charts

MANY MINI-UTILITIES: Disk Scanner finds bad disk sectors, **Key-Clicker** adds subtle sound as you type, **DOS-Killer** adds two tracks of space to your disks, **2-Track Cat** allows up to 210 file names per disk, **Program Splitter** makes room for hi-res pix with large Applesoft programs, **Text Imprinter** transfers text to the hi-res screen, **Onerr Tell Me** prints the appropriate error message but continues program execution, **Text Screen Formatter** converts text layouts into Print statements... plus much more Apple wizardry from the boys at Beagle Bros.

MORE TIPS ON DISK: Including fantastic programming tricks from Beagle Bros Tip Books 5, 6 and 7, plus programs from Tips/Tricks Chart #1.

TWO-LINERS TOO: From our customers around the world—and elsewhere. Little mind-blowers that will teach your old Apple some new tricks!

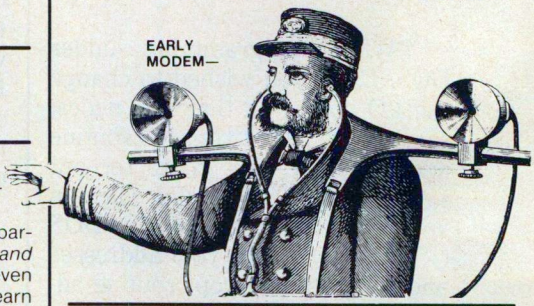
TIP DISK #1

100 TIP BOOK TIPS ON DISK
by BERT KERSEY

$20.00: Includes Peeks & Pokes Chart.

100 LISTABLE PROGRAMS from Beagle Bros Tip Books 1-4. Make your Apple do things it's never done! All 100 programs are LISTable and changeable for Apple experimentation.

COMMAND CHART INCLUDED: Free with each Tip Disk; an 11 x 17 poster of all Applesoft, Integer Basic & DOS Commands with Descriptions!



FLEX TYPE

(FORMERLY "FLEX TEXT")
VARIABLE-WIDTH HI-RES TEXT UTILITY
by MARK SIMONSEN

$29.50: Includes Peeks & Pokes Chart

PRINT VARIABLE-WIDTH TEXT on both hi-res screens with normal Applesoft commands (including HTAB 1-70). Normal, expanded & compressed text with no extra hardware. (70-column text requires a monochrome monitor, not a tv).

ADD GRAPHICS TO TEXT or add Text to hi-res graphics. Run your existing Applesoft programs under Flex Type control. Fast, easy to use, and Compatible with GPLE and Double-Take.

DOS TOOL KIT® font compatibility, or use the supplied Flex Type typefaces. Select up to 9 fonts with control-key commands. A text character editor lets you redesign any Apple text character.

FRAME-UP

FAST APPLE DISPLAY UTILITY
by TOM WEISHAAR

$29.50: Includes Peeks & Pokes Chart

PROFESSIONAL PRESENTATIONS: Turn your existing Hi-Res, Lo-Res and Text frames into attractive Apple "slide shows". *FAST* hi-res loads in 2½-seconds! Paddle or Keyboard-advance frames.

UNATTENDED SHOWS are optional, with each picture arranged and pre-programmed to display on the screen from 1 to 99 seconds. Custom **Text Screen Editor** lets you create black-and-white text "slides" and add type "live" from the keyboard during shows. Mail copies of presentations on disk to your friends and associates (or home to Mom!).

NEW!

GPLE

GLOBAL PROGRAM LINE EDITOR
by NEIL KONZEN

$49.95: Includes Peeks/Pokes Chart & Tip Book #7.

A CLASSIC APPLE PROGRAM EDITOR GPLE lets you edit Applesoft program lines *FAST* without awkward cursor-tracing and "escape editing".

INSERT & DELETE: GPLE works like a word processor for Applesoft program lines. You make changes instantly by jumping the cursor to the change point and inserting or deleting text. No need to trace to the end of a line before hitting Return.

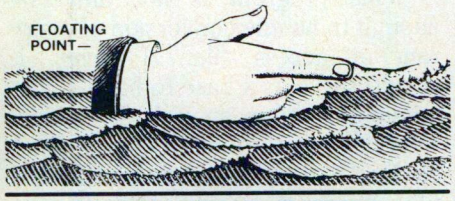
GLOBAL SEARCH & REPLACE: Find any word or variable in your programs, *FAST*. For example, find all lines containing a GOSUB, or edit or delete all lines with REM statements, or all occurrences of any variable. **Replace any variable**, word or character with any other. For example, change all X's to ABC's, or all "Horse" strings to "Cow".

80-COLUMN COMPATIBILITY: All edit & global features support **Apple IIe 80-column cards** and most 80-column cards on *any* Apple IIe, II+ or II.

DEFINABLE ESC FUNCTIONS: Define ESC plus any key to perform any task. For example, **ESC-1** can catalog drive 1, **ESC-L** can do a "HOME: LIST", **ESC-N** could type an entire subroutine... *Anything* you want, *whenever* you want.

GPLE DOS MOVER: Move DOS and GPLE to Language Card (or IIe upper 16K) for an EXTRA **10,000 Bytes** (10K) of programmable memory.

Plus APPLE TIP BOOK #7: Learn more about your Apple! Includes all new GPLE tips and tricks.



UTILITY CITY

21 PROGRAMMING UTILITIES
by BERT KERSEY

$29.50: Includes Peeks/Pokes Chart & Tip Book #3

LIST FORMATTER prints each program statement on a new line. Loops indented with printer Page Breaks. A great Applesoft program de-bugger.

MULTI-COLUMN CATALOGS, with or without sector and file codes. Organize your disk library.

INVISIBLE and trick catalog file names. Invisible functioning commands in Applesoft programs too.

MUCH MORE: 21 utilities, including auto-post Run-number & Date in programs, alphabetize/store info on disk, convert dec to hex or Int to FP, protect and append programs, dump text to printer...

LEARN PROGRAMMING: List-able programs and informative documentation. Includes Tip Book #3. Hours of good reading & Applesoft experiments.

ALPHA PLOT

HI-RES GRAPHICS/TEXT UTILITY
by BERT KERSEY and JACK CASSIDY

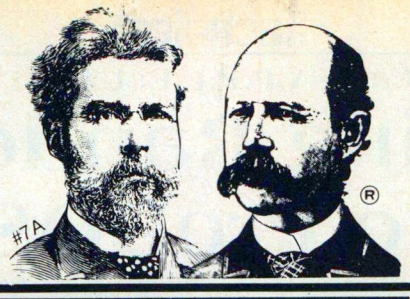
$39.50: Includes Peeks/Pokes Chart & Tip Book #4.

DRAW IN HI-RES on both Apple "pages" using easy keyboard commands OR paddles/joystick. Pre-view lines before plotting. Solid or mixed colors & Reverse (background-opposite) drawing. FAST one-keystroke circles, boxes & ellipses, filled or outlined. Add text for graphs & charts. All pix Save-able to disk, to be called from your Applesoft programs.

COMPRESS HI-RES DATA to 1/3 disk space (average) allowing more hi-res pictures per disk.

MANIPULATE IMAGES: Superimpose any two images, or RE-LOCATE any rectangular section of any drawing anywhere on either hi-res page.

HI-RES TYPE: Add text to your pictures with adjustable character-size and large-character color. Type *anywhere* with no Htab/Vtab limits. Type sideways too, for graphs. Includes Tip Book #4.



Beagle Bros Micro Software Inc.

4315 SIERRA VISTA / SAN DIEGO, CA 92103
619-296-6400

ALL BEAGLE DISKS ARE UNLOCKED, COPYABLE AND COMPATIBLE WITH APPLE II, II+ AND IIe.*

(Don't Settle for Less!)

* DISKQUIK requires Apple IIe.
"APPLE" is a Registered Trade Mark of You-Know-Who.

BEAGLE BASIC

APPLESOFT ENHANCER
by MARK SIMONSEN

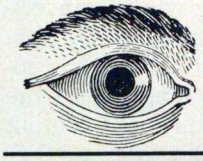
$34.95: Includes Peeks/Pokes Chart & Tip Book #6.
Requires Apple IIe (OR II/II+ with RAM Card).

RENAME ANY APPLESOFT COMMAND or Error Message to anything you want. For program clarification, encryption/protection or even foreign translation. Plus add optional NEW COMMANDS:

ELSE follows If-Then statements, like this:
IF X=2 THEN PRINT "YES": *ELSE* PRINT "NO"

HSCRN reads color of any hi-res dot for collision testing. **SWAP X,Y** exchanges 2 variables' values. New **TONE** command writes music with no messy pokes & calls. **SCRL** scrolls text in *either* direction. **TXT2** lets Text Page 2 act exactly like Page 1.

PLUS: GOTO & GOSUB may precede variables, as in "GOSUB FIX" or "GOTO 4+X". Escape-mode indicated by special ESC CURSOR. Replace awkward Graphics screen-switch pokes with 1-word commands. Change ctrl-G Beep to any tone. **INVERSE REMS** too! All GPLE compatible.



```
1 FOR S=768 TO 773: READ A:
POKE S,A: NEXT: POKE 232,0:
POKE 233,3: DATA 1,0,4,0,5,0
2 HGR2: FOR R=0 TO 192: ROT=R:
SCALE=96: XDRAW 1 AT 140,95:
SCALE=30: XDRAW 1 AT 140,95:
S=PEEK(49200): NEXT: RUN
```

PRONTO-DOS

HIGH-SPEED DOS / DOS-MOVE UTILITY
by TOM WEISHAAR

$29.50: Includes Peeks & Pokes Chart

TRIPLES THE SPEED of disk access and frees 10,000 bytes of extra memory by moving DOS.

Function	Normal	Pronto
BLOAD HI-RES IMAGE	10 sec.	3 sec.
LOAD 60-SECTOR PROGRAM	16 sec.	4 sec.
SAVE 60-SECTOR PROGRAM	24 sec.	9 sec.
BLOAD LANGUAGE CARD	13 sec.	4 sec.

(Text Files: No Change)

Boot the Pronto disk or your updated disks, created with the normal INIT command. Compatible with all DOS Commands, GPLE, Double-Take, DOS Boss, DiskQuik and almost all unprotected programs.

MOVE DOS to your Language Card, RAM Card, or standard Apple IIe upper 16K, freeing up **10,000 EXTRA BYTES** of memory for your programs.

15 EXTRA SECTORS per disk. Catalog Free-Space displayed every time you catalog a disk.

TYPE-COMMAND ("TYPE filename") prints contents of sequential Text Files on screen or printer.

NEW!

DISKQUIK

DISK DRIVE EMULATOR
by HARRY BRUCE and GENE HITE

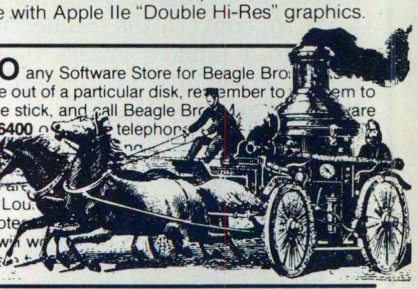
$29.50: Includes Peeks & Pokes Chart
Requires Apple IIe with Extended 80-column Card.

ACTS LIKE A DISK DRIVE in Slot 3, but much faster, quieter, more reliable and $350+ cheaper! Enjoy the benefits of a 2nd (or 3rd or 4th...) drive at less than 1/10th the price. Catalogs normally with "CATALOG, S3" command. Load & Save any kind of files into RAM with normal DOS commands.

SILENT AND FAST: Since no moving parts are involved, DiskQuik operates silently and at super-high speeds. See it to believe it. Your Apple IIe's Extended 80-column Card (required) can hold about half the amount of data as a 5¼" floppy disk!

MANY USES: For example, auto-load often-used files like FID etc., etc., into RAM when you boot up, so they are always available when you need them. Copy files from RAM onto disk and vice versa, just as if a disk drive were connected to slot #3.

FRIENDLY & COMPATIBLE with 80-column display, GPLE, ProntoDOS, and all normal Applesoft and DOS commands and procedures. Will not interfere with Apple IIe "Double Hi-Res" graphics.



GOTO any Software Store for Beagle Bros. If they are out of a particular disk, remember to [illegible] to get on the stick, and call Beagle Bros at **619-296-6400** on [illegible] telephone. Distribut[illegible] every[illegible] our dis[illegible] to Uncle Lou[illegible] are unprote[illegible] floppies wi[illegible] everywhe[illegible]

DOUBLE-TAKE

2-WAY-SCROLL/MULTIPLE UTILITY
by MARK SIMONSEN

$34.95: Includes Peeks/Pokes AND Tips/Tricks Charts.

2-WAY SCROLLING: Listings & Catalogs scroll Up AND Down, making file names and program lines much easier to access. Change the Catalog or List scroll-direction at will, with Apple's Arrow keys.

80-COLUMN COMPATIBLE: All features support IIe and most other 80-column cards.

BETTER LIST FORMAT: Each program statement lists on a new line for *FAST* program tracing & de-bugging. Printer-compatible; any column-width.

VARIABLE-DISPLAY: Displays all of a program's strings and variables with current values.

CROSS-REFERENCE: Sorts and displays line numbers where each variable & string appears.

AUTO-LINE-NUMBER, Hex/Dec Converter, better Renumber/Append, Program Stats, Change Cursor, Space-On-Disk. GPLE/Pronto compatible.

☐ Alpha Plot	$39.50	☐ Frame-Up	$29.50
☐ Apple Mechanic	29.50	☐ GPLE	49.95
☐ A.M.Typefaces	20.00	☐ ProntoDOS	29.50
☐ Beagle Bag	29.50	☐ Silicon Salad	24.95
☐ Beagle BASIC	34.95	☐ Tip Disk #1	20.00
☐ DiskQuik	29.50	☐ Utility City	29.50
☐ DOS Boss	24.00	☐	
☐ Double-Take	34.95	☐ ADD ME to mailing list.	
☐ Flex Type	29.50	☐ ALREADY ON mail list.	

AT YOUR APPLE DEALER NOW!
Or order directly from Beagle Bros—

VISA · master charge

Visa/MasterCard or COD, **call TOLL-FREE**
Orders only / ALL 50 STATES / 24 Hours a Day

1-800-227-3800 ext. 1607

OR mail U.S.Check, Money-Order or Visa/MC#
to **BEAGLE BROS**, 7th Floor
4315 SIERRA VISTA, SAN DIEGO, CA 92103

Add $1.50 First Class Shipping. Any-Size Order.
Overseas add $4.00. COD add $3.00. California add 6%
ALL ORDERS SHIPPED IMMEDIATELY.

CLIP COUPON OR USE SEPARATE SHEET

A Call –A.P.P.L.E. UTILITY:

Converting Random Files to ProDOS

ken kashmarek

PRODOS CONVERT does not handle conversion of DOS 3.3 random access files. Therefore, it is necessary to convert DOS 3.3 random access files to sequential files before they can be converted to ProDOS. Once the file is moved to ProDOS, it is then necessary to convert the sequential file back to a random access file. Random/Sequential File Converter will help get this job done.

The program may be run under DOS 3.3 or ProDOS. Once the program has been entered and saved, it may be moved to ProDOS by using the ProDOS CONVERT program. It will then run as is under ProDOS.

Assuming that you plan to convert a DOS 3.3 random access file to ProDOS, the first step is to run Random Sequential File Converter under DOS 3.3 and convert your random access file to a sequential file. The program will prompt you for the input filename (random access) and output file name (sequential). You must then supply the number of fields per random record and the random access file length.

Each field in a random access record is separated by carriage return characters ($8D). This program assumes the same number of fields in each record of the file. The data in the random access file is read beginning with record #1 (not record #0, which is the first record in a random access file). In most cases, record #0 does not contain a normal data record. Typically, it contains a count of the number of records in the file. Some applications will also place the record length and/or the number of fields per record in record #0. Since record #0 usually does not contain the same number of fields as the other records, it is not processed. It may be necessary for you to re-create record #0 after converting the file to ProDOS, or modify the program to copy record #0 separately.

To facilitate the conversion, a version of the INPUT ANYTHING routine is installed to read the fields (see *Call –A.P.P.L.E. In Depth 1:* All About Applesoft). This is done with GOSUB 530. This very short routine is located in page 3. This takes care of records that have special characters that Applesoft does not allow during string input.

This program will not handle sparse random access files. A sparse file contains empty or nonexistent records within the file. If your random access file has empty records within the file, those records must contain a carriage return ($8D) character for each field in the record. Random to sequential conversion terminates when end of file (PEEK(222) = 5) is reached.

Once the random file is converted to sequential, then you must use ProDOS CONVERT to convert the sequential text file from DOS 3.3 to ProDOS. After the file is converted to ProDOS, rerun Random Sequential File Converter to create a random access file under ProDOS.

At this point, you will be converting the ProDOS sequential file version of your data to a ProDOS random access file. You must supply the input pathname (sequential), output pathname (random access), number of fields per record, and record length. When ProDOS creates a random access file, it stores the record length with the name of the file. Once the file is created, it is not necessary to specify record length. The sequential file is then read, and the random access records are created beginning with record #1. The number of records is stored in record #0 at the end of the conversion.

If record #0 in your DOS 3.3 random access file contained other information, you may now recreate that information at this time under ProDOS. You may now run your application (if it was converted from DOS 3.3), or write a quick special program to read and display the contents of the file. This is usually necessary to verify that the data transported correctly from DOS 3.3 to ProDOS.

If your file is very large, then you must have two disk drives for the conversion. This will also be necessary for the ProDOS CONVERT routine to work. Under ProDOS, separate disk drives are identified in the pathname by the volume name of the diskette. Under DOS 3.3, you will have to add the drive number to line 420.

This program is not universal enough to handle conversion of every random access file. However, it should serve as a base to handle conversion of your special file needs. Very few of the statements depend on ProDOS. If your random access file is sparse, then you must add code to handle that situation. In this case, when creating the sequential file, it will be necessary to write out the random access record number as well. This record number would then be used under ProDOS to replace the random record at the proper position in the file.

If your random access file has records with differing numbers of fields, then special care must be used. Information concerning the number of fields must also be added to the sequential file so the random access file can be properly reconstructed under ProDOS.

A special note: Under DOS 3.3, location 222 contains a 5 when end of data is reached in a random access record or when the end of file is reached. Under ProDOS, location 222 contains a 2 indicating out of data when a zero byte is detected in a random access record. The 2 indicates a ProDOS parameter range error. In this case, you are trying to access a field beyond the fields in a random access record. This means the field parameter is too large, giving the range error (even if a field parameter was not specified explicitly).

```
]LIST

100  REM

CONVERT.RANDOM BY KEN KASHMAREK

        CALL -A.P.P.L.E.


110  REM

CONVERTS DOS 3.3 RANDOM FILES
TO PRODOS FORMAT

120  DIM A$(32),F$(2):F$(1) = "SE
     QUENTIAL TO RANDOM":F$(2) =
     "RANDOM TO SEQUENTIAL": GOSUB
     530: POKE 1014,0: POKE 1015,
     3
130  TEXT : HOME : FOR I = 1 TO 4
     1: PRINT "*";: NEXT : HTAB 4
     0: PRINT "**";
140  HTAB 7: PRINT "RANDOM ACCESS
      FILE CONVERTER";
150  HTAB 40: PRINT "**";: HTAB 4
     0: FOR I = 1 TO 41: PRINT "*
     ";: NEXT
160 ER = 5:PA$ = "FILE":EX$ = "DO
     S 3.3": IF  PEEK (977) = 0 AND
      PEEK (978) = 190 THEN ER =
     2:PA$ = "PATH":EX$ = "PRODOS
     "
170  PRINT : PRINT "YOU ARE RUNNI
     NG UNDER ";: INVERSE : PRINT
     EX$: NORMAL
180  PRINT : PRINT "   1. ";F$(1)
     : PRINT : PRINT "   2. ";F$(
     2): PRINT : PRINT "SELECT ";
     : GET E$: PRINT : IF E$ =  CHR$
     (13) THEN  PRINT : PRINT "GO
     ODBYE.": END
190  IF E$ <  > "1" AND E$ <  > "
     2" THEN  PRINT  CHR$ (7): GOTO
     130
200 EX =  VAL (E$):D$ =  CHR$ (4)

210  VTAB 6: CALL  - 958: PRINT :
      PRINT "CONVERT ";F$(EX)
220  REM PROMPT FOR FILE INFO
230  PRINT : PRINT "INPUT ";PA$;"
     NAME ": PRINT : HTAB 4: INPUT
     "";IP$
240  PRINT : PRINT "OUTPUT ";PA$;
     "NAME ": PRINT : HTAB 4: INPUT
     "";OP$
250  PRINT : INPUT "# OF FIELDS P
     ER RANDOM RECORD ";FP
260  IF  NOT (EX$ = "PRODOS" AND
     EX = 2) THEN  INPUT "RANDOM
     FILE RECORD LENGTH     ";RL
270  ON EX GOTO 280,390
280  REM SEQUENTIAL TO RANDOM
290  PRINT D$"OPEN";IP$
300  PRINT D$"OPEN";OP$;",L";RL
310 R = 0: ONERR  GOTO 370
320  PRINT D$"READ";IP$
330  FOR I = 1 TO FP: &  INPUT A$
     (I): NEXT
340 R = R + 1: PRINT D$"WRITE";OP
     $;",R";R
350  FOR I = 1 TO FP: PRINT A$(I)
     : NEXT
360  GOTO 320
370  IF  PEEK (222) <  > 5 THEN 5
     00
380  PRINT D$"WRITE";OP$;",R0": PRINT
     R: GOTO 510
390  REM RANDOM TO SEQUENTIAL
400  IF EX$ = "PRODOS" THEN  PRINT
     D$"OPEN";IP$: GOTO 420
410  PRINT D$"OPEN";IP$;",L";RL
420  PRINT D$"OPEN";OP$
430 R = 0: ONERR  GOTO 490
440 R = R + 1: PRINT D$"READ";IP$
     ;",R";R
450  FOR I = 1 TO FP: &  INPUT A$
     (I): NEXT
460  PRINT D$"WRITE";OP$
470  FOR I = 1 TO FP: PRINT A$(I)
     : NEXT
480  GOTO 440
490  IF  PEEK (222) = ER THEN R =
     R - 1: GOTO 510
500  PRINT "ERROR "; PEEK (222);"
      AT LINE "; PEEK (218) +  PEEK
     (219) * 256: END
510  POKE 216,0: PRINT D$"CLOSE"
520  PRINT : PRINT R;" RECORDS PR
     OCESSED.": PRINT : PRINT "FI
     NISHED.";: END
530  GOTO 550
540 H$ = H$ + "ND9C6G": FOR I = 1
      TO  LEN (H$): POKE 511 + I,
      ASC ( MID$ (H$,I,1)) + 128:
      NEXT : POKE 72,0: CALL  - 1
     44: RETURN
550 H$ = "0300:A9 84 20 C0 DE 20
     E3 DF 20 6C DD A2 80 86 33 2
     0 6A FD 86 06 20 39 D5 A5 06
      20 52 E4 A2 00 A0 02 20 E2
     E5 A0 00 A5 06 91 83 C8 A5 6
     F 91 83 C8 A5 70 91 83 60 ":
      GOSUB 540
560  RETURN
```

A Call –A.P.P.L.E. TUTORIAL:

Indexing Files

scott g smith

THERE comes a time in a data file's history when it becomes too large to be kept entirely in memory. Perhaps the amount of data has grown, or perhaps the program that maintains it has. At that time performance is compromised or impossible. A decision is made to store the data in a random access file, each logical record becoming a physical record. Accessing records based on one or more of the records' fields entails searching the entire file. But the file has grown so large that searching the records sequentially is too slow. What are the options for improving search/access times?

I'd like to suggest the idea of maintaining an index to the file. The index should consist of one field that uniquely identifies its record (i.e., the key field). In the case of a mailing list, that field might be the last name (to uniquely identify a record, maybe the first name or initial might have to be appended to the last name, like SMITH-S, SMITH-SCOTT, or SMITH-S-C). In the case of a file of books, perhaps the key field would be the title (though to uniquely identify a title, the author's initials may also be necessary). This index will itself be stored in a file (sequential, for reasons that will be clear later), will be read in by any program that manipulates the file's data, and written back to disk if any changes are made to it.

An index offers the following advantages. Memory use is less; the index is considerably smaller than the entire file. Once a record is located by searching the index, only that one record need be read into memory. Searching the index in memory is also quicker than searching a file on disk. Ordering the file by its key field is far quicker with the index than by reading the file and ordering it. And if necessary, the file can be ordered by secondary fields as well, with similar constructs.

A Simple Index

The simplest index could be implemented as a single subscripted string variable where the subscript is also the record number (Figure 1).

```
I  A$(I)

1  PASCAL-BLAISE
2  BABBAGE-CHARLES
3  VON NEUMANN-JOHN
4  HOLLERITH-HERMAN
5  NAPIER-JOHN
```

Figure 1. A simple index.

Thus the data pertaining to PASCAL-BLAISE is kept in record 1, for BABBAGE-CHARLES in record 2, and so on. As the file grows, so does the index. Though searching this index is a vast improvement over a sequential search of the file, there are three drawbacks to this index. A name would be located in the index by a sequential search; this is a slow search technique for a large index. Secondly, to order the file would require sorting the index. But the index can't be sorted if the sort changes the subscripts — they must be preserved as record numbers. And thirdly, if a record is deleted, a null entry must replace the other, and the deleted record number should itself be saved (so deleted records can be re-used). A second subscripted variable, this one real/integer and stored with the index (a sequential file accommodates them both easier than a random access file), could hold these deleted record numbers. That is, if ER is the count of empty records, then ER(1), ER(2), . . . , ER(ER) would be the deleted record numbers. When a new record is added, the value of ER should be checked, If non-zero, then the record number assigned it should be ER(ER) and ER decremented. The array ER(1), ER(2), . . . , ER (ER) is thus seen to be a LIFO (last in/first out) queue.

A Simple Index With Position Pointers

One way of sorting the simple index in Figure 1 is by using "position Pointers." If LR is the last record in the file (not necessarily the number of records because ER of them are deleted), then P(1), P(2), . . . , P(LR) are position pointers. P(I) would hold the subscript of the index entry in the I th position. The sorting method compares entries indexed by the P array and switches the P values, not the index entries themselves. For instance, instead of the normal "selection sort" in Figure 2a, the modifications indicated in Figure 2b would allow the index to be sorted without changing the index's subscripts (P(1), P(2), . . . , P(LR) would be initialized to 1, 2, . . . , LR preceding the sort). Figure 3 illustrates the index from

```
20  FOR I = 1 TO N -
      1: FOR J = I +
      1 TO N
30  IF A$(I) > A$(J)
      THEN S$ = A$(I):A$(I)
      = A$(J):A$(J)
      = S$
40  NEXT J,I
```

Figure 2a. Selection Sort.

```
20  FOR I = 1 TO N -
      1: FOR J = I +
      1 TO N
30  IF A$(P(I)) > A$(P(J))
      THEN S = P(I):P(I)
      = P(J):P(J) =
      S
40  NEXT J,I
```

Figure 2b. Selection Sort (with position pointers).

```
I   P(I) A$(I)

1    2 PASCAL-BLAISE
2    4 BABBAGE-CHARLES
3    5 VON NEUMANN-JOHN
4    1 HOLLERITH-HERMAN
5    3 NAPIER-JOHN
```

Figure 3. A simple index (with position pointers).

Figure 1 with the position pointers sorted. Note that as I takes the values from 1 to LR, that A$(P(I)) lists the index alphabetically. A problem does exist with the index entries for empty records however. Should they be given values so they get sorted to the beginning or to the end of the index? Regardless, avoiding these entries is a necessary part of any code that prints the index or the file in order.

Index as a Linked List

Making the index a linked list eliminates the null entry problem. A linked list can be pictured as in Figure 4. Each entry consists of two fields, the data and the pointer to the next (in this case, alphabetically speaking) entry in the list. The head of the list points to (i.e., is the subscript of) the first ordered entry in the index. Each entry's pointer contains the subscript of the entry that follows it. The end of the list necessarily has a null pointer. Figure 5a indicates that L(0) points to the list's head, and that L(1) through L(5) are the links (pointers). If a record is deleted (Figure 5b), by redefining a single link, the index stays ordered. If a record is added to the original (Figure 5c), two links must be redefined to keep the index sorted. In both these cases, handling the special cases (first entry and last entry) requires care in coding. In any case, a list of deleted record numbers ER(1), ER(2), . . . , ER(ER) must still be kept as before. Searching the index requires traversing the entries by following the pointer values. The problem with this is in search time. Each search entails following the head of the list to the appropriate position in the list; like a sequential search, on the average half the entries must be looked at for any given search. Because searching a linked list is fundamental to the addition and deletion routines (to locate the links that need redefinition), a slow sequential-like search is a drawback. Though backward pointers could be kept to improve search speed (i.e., search backwards if the entry begins with a letter after M), it would require twice the memory (another array) and twice the complexity in the update code.

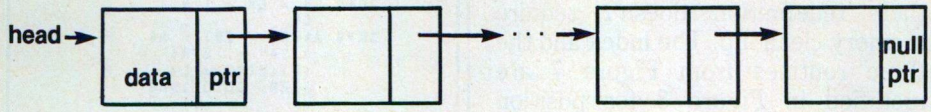


Figure 4. A linked list.

Position Pointers, Binary Search/Insertion/Deletion

Though a linked list solves the null entry problem, search time is poor. But there is a way to keep the index sorted, the search times minimized, and the nuli entries deleted. This scheme requires keeping only valid entries in the index, and keeping it sorted at all times. Because the subscripts will change in this data structure as the entries are rearranged, they can no longer double as record numbers. The record numbers will have to be kept another way. One way to do this is to store the record numbers in a separate array (Figure 6a). However, this means manipulating two arrays instead of one for each update. A way around this is to make the record number into a string and append this string to each entry. This could be done in any of three ways so that the record number is easily accessible. Firstly, make the record string a constant length (making a record string of length 4 could be done with

```
R$ = STR$(R):IF LEN(R$)<4 THEN
FOR I=LEN(R$)+1 TO
4:R$=R$+" ":NEXT I
```

and concatenate this string and the key field e.g., "4 PASCAL-BLAISE").

Secondly, append any length record string to an identification field of constant length (by either truncating the key field or padding it with blanks). Or thirdly, append the record string to the key field, separating them with a special character (Figure 6b). The

```
I   L(I) A$(I)

1    3 PASCAL-BLAISE
2    5 BABBAGE-CHARLES
3    0 VON NEUMANN-JOHN
4    0 * (optional)
5    1 NAPIER-JOHN

head = L(0) = 2 ER = 1 ER(1) = 4
```

Figure 5b. Linked list with a deletion.

```
I   R(I) A$(I)

1    2 BABBAGE-CHARLES
2    4 HOLLERITH-HERMAN
3    5 NAPIER-JOHN
4    1 PASCAL-BLAISE
5    3 VON NEUMANN-JOHN
```

Figure 6a. Sorted index with separate record numbers.

```
I   L(I) A$(I)

1    3 PASCAL-BLAISE
2    4 BABBAGE-CHARLES
3    0 VON NEUMANN-JOHN
4    5 HOLLERITH-HERMAN
5    1 NAPIER-JOHN

head = L(0) = 2
```

Figure 5a. Index as linked list.

```
I   L(I) A$(I)

1    3 PASCAL-BLAISE
2    4 BABBAGE-CHARLES
3    0 VON NEUMANN-JOHN
4    6 HOLLERITH-HERMAN
5    1 NAPIER-JOHN
6    5 JOBS-STEVE

head = L(0) = 2
```

Figure 5c. Linked list with an addition.

```
I   A$(I)

1  BABBAGE-CHARLES*2
2  HOLLERITH-HERMAN*4
3  NAPIER-JOHN*5
4  PASCAL-BLAISE*1
5  VON NEUMANN-JOHN*3
```

Figure 6b. Sorted index with record numbers appended.

reason to keep the index sorted is to allow a binary search, and concomitantly, quick insertions and deletions (as with the linked list, the search determines where an insertion or deletion will occur). A list of empty records must still be kept. A binary search for such an index is coded in Figure 7a. Part 7b uses part 7a to insert a new entry (the entries from the place of insertion to the end of the index must be shifted down one subscript value). Part 7c illustrates a deletion (entries must be moved up one subscript from the bottom of the index to the place of deletion). Note

```
1000  REM
BINARY SEARCH FOR
A$ IN A$(1)
.. A$(LR-ER)
1010 LO = 0:HI = LR
     - ER + 1:X1 =
     0
1020 X =  INT ((LO +
     HI) / 2): IF X
     = X1 THEN X1 =
     0: RETURN
1030  IF A$ > MID$
     (A$(X),1, LEN
     (A$)) THEN LO
     = X:X1 = X: GOTO
     1020
1040  IF A$ < MID$
     (A$(X),1, LEN
     (A$)) THEN HI
     = X:X1 = X: GOTO
     1020
1050 X1 = X: RETURN
```

Figure 7a. Binary Search.

```
2000  REM
BINARY INSERTION
2010  INPUT "NAME TO
      INSERT? [OR RET]
      ";A$
2020  IF A$ = "" THEN
      2010
2025  IF LR = 0 THEN
      X = 1: GOTO 2060
2030  GOSUB 1000:X =
      X + 1
2040  FOR I = LR - ER
      + 1 TO X + 1 STEP
       - 1:A$(I) = A$(I
      - 1): NEXT
2050  IF ER <> 0 THEN
      R = ER(ER):ER
      = ER - 1: GOTO
      2070
2060 LR = LR + 1:R =
      LR
2070 A$(X) = A$ + "*"
      +  STR$ (R): GOTO
      2000
```

Figure 7b.
Binary search and insertion.

that if the search subroutine returns with X1 equal to 0, no matching entry has been found; and if X1 is non-zero, X is the subscript of the matching index entry. If redefining many entry strings (as required by the routines in Figures 7b and 7c) slows performance, position pointers can be used so that only the values of the position pointers are redefined — the position pointers are real/integer arrays, and their redefinition doesn't require memory clean-up. The index and the three routines from Figure 7 are modified in Figure 8 for position pointers.

```
3000  REM
BINARY DELETION
3010  INPUT "NAME TO
      DELETE? [OR RET]
      ";A$
3020  IF A$ = "" THEN
      3010
3030  GOSUB 1000: IF
      X1 = 0 THEN  PRINT
      "CAN'T FIND "A$"
      TO DELETE": PRINT
      : GOTO 3000
3040  FOR I = 1 TO
      LEN (A$(X)): IF
       MID$ (A$(X),I,1)
      <> "*" THEN
       NEXT
3045 ER = ER + 1:ER(ER)
      =  VAL ( MID$
      (A$(X),I + 1,
      LEN (A$(X)) -
      I))
3050  FOR I = X TO LR
      - ER:A$(I) = A$(I
      + 1): NEXT
3060  GOTO 3000
```

Figure 7c. Binary search and deletion.

Indices for Secondary Attributes

Though the binary search with insertions and deletions is a very efficient way of maintaining an index, it must be remembered that this index represents just one field of each record, the key field. If it is desirable to search the file on a second field as well, a similar index would be kept for this attribute. If the actual index entries of this second field would deplete too much memory, just the values of some position pointer, perhaps Q(1), Q(2), . . . , Q(LR–ER) could be kept in memory. Each comparison for this pointer's binary search would require a disk access (to get the secondary field from the record), but the file access would be minimized (because the binary search is efficient). The secondary attribute's position pointers can be changed in memory and re-written to the index file at the program's completion, just like the primary index.

I	P(I)	A$(I)
1	3	PASCAL-BLAISE*1
2	4	NAPIER-JOHN*5
3	2	BABBAGE-CHARLES*2
4	1	HOLLERITH-HERMAN*4
5	5	VON NEUMANN-JOHN*3

Figure 8a.
Index sorted by position pointers.

```
1000  REM
BINARY SEARCH FOR
A$ IN A$(P(1))
.. A$(P(LR-ER))
1010 LO = 0:HI = LR
     - ER + 1:X1 =
     0
1020 X =  INT ((LO +
     HI) / 2): IF X
     = X1 THEN X1 =
     0: RETURN
1030  IF A$ > MID$
     (A$(P(X)),1, LEN
     (A$)) THEN LO
     = X:X1 = X: GOTO
     1020
1040  IF A$ < MID$
     (A$(P(X)),1, LEN
     (A$)) THEN HI
     = X:X1 = X: GOTO
     1020
1050 X1 = X: RETURN
```

Figure 8b.
Binary search (with position pointers).

```
2000  REM
BINARY INSERTION
2010  INPUT "NAME TO
      INSERT? [OR RET]
      ";A$
2020  IF A$ = "" THEN
      2010
2025  IF LR = 0 THEN
      X = 1: GOTO 2060
2030  GOSUB 1000:X =
      X + 1
2040  FOR I = LR - ER
      + 1 TO X + 1 STEP
       - 1:P(I) = P(I
      - 1): NEXT
2050  IF ER <> 0 THEN
      R = ER(ER):ER
      = ER - 1: GOTO
      2070
2060 LR = LR + 1:R =
      LR
2070 A$(LR - ER) = A$
      + "*" +  STR$
      (R):P(X) = LR
      - ER: GOTO 2000
```

Figure 8c.
Binary search and insertion
(with position pointers).

```
3000  REM
BINARY DELETION
3010  INPUT "NAME TO
      DELETE? [OR RET]
      ";A$
3020  IF A$ = "" THEN
      3010
3030  GOSUB 1000: IF
      X1 = 0 THEN  PRINT
      "CAN'T FIND "A$"
      TO DELETE": PRINT
      : GOTO 3000
3040  FOR I = 1 TO
      LEN (A$(P(X))):
      IF  MID$ (A$(P(X)),I,1
      ) <> "*" THEN
       NEXT
3045 ER = ER + 1:ER(ER)
      =  VAL ( MID$
      (A$(P(X)),I +
      1, LEN (A$(P(X)))
      - I))
3050  FOR I = X TO LR
      - ER:P(I) = P(I
      + 1): NEXT
3060  GOTO 3000
```

Figure 8d.
Binary search and deletion
(with position pointers).

Discover the Magic of your Apple!

the APPLE'S APPRENTICE™

The FIRST Magazine for Kids with Apples!

The Apple's Apprentice is the first magazine dedicated to kids with Apple computers at home or school. The Apple's Apprentice can help kids learn about computers and show them how to get the most out of their Apple, as well as what the rest of the world is doing with computers.

In Apple's Apprentice, you'll find a fascinating variety of informative articles and features like our series on the inside scenes of computer manufacturing, from the design of the chips to the building of printers.

There are also tutorials like **MAGIGRAPHICS, THE SOURCEROR'S APPRENTICE,** and **APPRENTICE BASIC. PLUS:** Interviews with famous people, kids making money with their computers, reviews of computer hardware and software, games, contests, puzzles, posters, cartoons, stories, seasonal projects and more!

Subscribe NOW to The Apple's Apprentice for only $18.00 (one-year) by sending in the money-saving coupon at right (Save $12.00 off the $30.00 single-copy cost and $6.00 off the regular $24.00 subscription rate)!

The Apple's Apprentice •Box 582-AA •Santee, CA 92071 •(619) 562-7785

GIFT SUBSCRIPTION COUPON

PLEASE SEND MY ONE-YEAR GIFT SUBSCRIPTION FOR $18.00 TO:

Name _____

Street _____

City _____ State _____ Zip _____

SEND GIFT COURTESY OF:

My Name _____

Street _____

City _____ State _____ Zip _____

☐ **Please start MY OWN subscription — also $18.00**

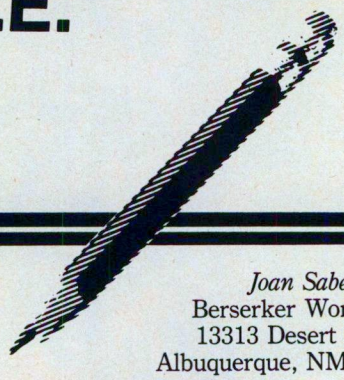
☐ Check ☐ Money Order ☐ MasterCard ☐ Visa

Card No. _____ expires: _____

Signature _____

Please list additional gifts on a separate sheet of paper.

WRITE —A.P.P.L.E.



Dear Mr. Golding:

THE MPI printer (88g or 99g) is one of a few printers made in the U.S.A. I hope it will gain a good reputation in this competitive world. Its mechanical design actually can do more than it was designed for. For example, although it does not provide a half line feed function, one can use the graphic printing to partly achieve this function and use it for printing superscript and subscript from a word processor program which is capable of scanning and printing in the order of superscripts/main line/subscripts within a line.

In Wordstar, patch the "half line feed–PSHALF location 06A8" to 04 0D 1B 17 34 to make a four dot vertical advance. There is a problem of graphic character printing I am not able to solve. MPI offers no help. The MPI printer has only one set of fonts. Its AP–PAK software, like many other softwares, can make graphic character printing but cannot print a graphic character in the middle of a line. This is not a problem for many other printers, but for some reason, MPI printers issue a carriage return whenever graphic printing is initiated. This eliminates the flexibility to start graphic character printing at any horizontal position. I don't know if this problem is coming from the program in the ROM chip, or in the logic circuitry. I wonder if anyone has contemplated this problem and could offer any solution?

P. Leung
15 Woodland Rd.
Highland Mills, N.Y. 10930

Dear Val,

Trying to run DOS 3.3 programs under ProDOS may involve some difficulties. A common problem involves incorporating machine language programs with a BASIC program.

Under DOS 3.3, a common practice is to move HIMEM, whose initial value is $9600, down to make room for machine language routines between the new HIMEM and the DOS file buffers. As long as one performs no file processing, this will probably work under ProDOS. In moving HIMEM, there is one "chiseled in stone" rule that HIMEM must be a multiple of 256, i.e., values like $8534 or $9031 are not legal. The last two hex digits must be zero.

However, if one performs file processing, odds are that the machine language code will be wiped out because HIMEM is not located 1K below the machine language code.

There is a clean fix to this problem. Write a short machine language subroutine that loads the A register with the number of pages needed for the code then JSR to GETBUF at $BEF5 (48885). If carry is returned clear, the A register will contain the high byte of the address into which code can be "safely" loaded and HIMEM has been moved down accordingly. Now the code can be BLOADed. This must be done in the order described, otherwise, the BLOAD may fail.

Notice that I said "safely." As long as one avoids the JSR to FREEBUFR at $BEF8 (48888), everything should be ok. If FREEBUFR is called, it doesn't care what has happened and returns with HIMEM set to $9600 every time. All of the GETBUF work has been wiped out.

There is an easy fix to this problem. FREEBUFR uses the contents of $BEFB (48891) to establish the final HIMEM. Its default value is $96 which is loaded into $BEFB when BASIC.SYSTEM is loaded. Therefore, set $BEFB to result in a HIMEM 1K below the desired "protected" area and FREBUFR will behave itself.

One final note. If one wants truly protected memory, then adjust the bit map starting at $BF58 (48984).

Cecil Fretwell
2605 Highview Avenue
Waterloo, IO 50702

Dear Jim Hockenhull,

I called about using a VHS recorder with an Apple. The method I used connects the Apple to the VCR input, the VCR output to the *monitor.* The VCR is set at low speed record with the *camera* input selected. The resulting recording will only play back on a monitor. The results were quite acceptable for my purposes.

Joan Saberhagen
Berserker Works Ltd
13313 Desert Flower
Albuquerque, NM 87111

Andromeda Softswitches

Dear Val,

Gentlemen,

I have been using an early release of Apple PIE (from Programma) for several years and have been quite pleased with its performance. Since I had expanded my Apple system to include an Andromeda 16K memory card, I had wanted to utilize the extra memory space. To do this, I wrote a "swap" program to exchange Format and PIE as required in the RAM card, reducing the several seconds of disk time to fractions of seconds. However, with the recent purchase of the Viewmax 80 column card, I was no longer satisfied with the 40 column operation. Due to my past satisfaction with the old version, and favorable reviews of PIE 2.2, I decided to purchase that version. You can imagine my disappointment when I loaded the system and found that I could not run Format in the 64K configuration. A check of system operation on a friend's system, configured with the Microsoft 16K card confirmed that it was the Andromeda card causing my problem. The purpose of this letter is to advise you of my corrective action, for possible assistance to future purchasers not as lucky as I at finding a solution.

The root problem with the Andromeda card is that the "soft switch" for write enabling the RAMcard memory is accessed only by reads to the latch addresses, whereas the Microsoft card can be read or written to. Where PIE accesses the alternate bank on the RAM card, or any other switching of banks, writes and reads are both used. I did not try changing the writes to reads as I assumed that the author probably already determined that the register contents had to be preserved else he would have read the latch. I did try using the BIT instruction, but between calls I must have altered some meaningful status information, and my only attempt failed. Having peaked at my frustration

level, and the time frame being over New Year's holiday, I assumed calls to any group would be unsuccessful, and resorted to hardware modification. While this fix may represent only one solution (maybe Hayden has a patch?), it does fix the read/write problem for this program, and possibly others.

> Pin 10 to U13 is the read/write input to the write enable logic. By removing IC U13, bending pin 10 outward, wire wrapping a connection to pin 10, reinserting the chip, and connecting the other end of the wire to a suitable pull up resistor (1K), the write enable circuit is now accessed by either a read or write operation.

I did not try letting the input to U13 float and its possible internal pull up circuitry may make my pull up resistor unnecessary, however, it is my opinion that each input to a used gate should be tied to the desired logic level.

I hope this information is useful and I wish to thank both Hayden and Andromeda for providing unprotected software and circuit diagrams, respectively, to allow users to trouble shoot these type of problems. Although in this case my solution was hardware, in another situation, a simple patch may have been equally effective.

John Wallace
2832 42 Street
Two Rivers, WI 54241

Inputting TASC

Dear Val:

I read with interest the "Exploiting TASC" article in the December 1983 issue of *Call -A.P.P.L.E.* by Sam F. Vass and found much valuable information in it. Let me share a trick which I found after hours of searching for a simple way to simulate LINE INPUT using TASC (The Applesoft Compiler by Microsoft). The Input Nearly Anything is excellent for Applesoft but does not work using the TASC compiler because it is difficult to pass parameters in a CALL statement using the compiler.

My solution was to modify the INPUT routine in TASC so that it uses only a carriage return as a delimiter, thereby allowing commas and semicolons within the string being input. To do this, simply POKE locations 4779 (decimal) through 4782 with NOP's before using the INPUT statements. I would strongly suggest restoring these locations after the LINE INPUT use is no longer desired because they appear to be used for other purposes by TASC and can cause unpredictable results.

Compile and run the following demonstration program to illustrate the use of the modified INPUT statement:

```
100 REM LINE INPUT USING
    TASC DEMO
110 REM ROBERT JONES
120 REM DECEMBER 1983
190 REM CONVERT INPUT TO
    LINE INPUT
200 POKE 4779,234: POKE
    4780,234:POKE 4781,234:POKE
    4782,234:REM NOP's
205 REM NOW USE IT
210 INPUT"ENTER TEXT WITH
    COMMAS ";T$
220 PRINT T$
225 REM
227 REM RESTORE NORMAL
    VALUES INTO TASC
230 POKE 4779,160:POKE
    4780,44:POKE
    4781,132K:POKE 4782,22:REM
    RESTORETR
240 END
```

The above segment should solve the LINE INPUT problem in a quick and easy manner. I hope that your readers find it useful. Remember that this method will not work with the Applesoft interpreter. If can only be used in a program to be compiled using TASC at the default locations.

Robert Jones
7 Lexington Road
Saratoga Springs, NY 12866

Irish Apples Smiling!

Dear Mr. Golding,

I am writing in answer to your request for comments on Mr. Richard King's letter concerning the differences between "standard" Apple *IIe*'s and the "Irish Apple" (Write -A.P.P.L.E. December '83). I have recently changed from a][+ to an Irish Apple and despite the placement of the auxiliary slot in front of slot 3, I still have an extra slot available due to the fact that I no longer need a PAL card in slot 7 just to get the colors which were standard on domestic][+'s. This is a great leap forward, no doubt warmly welcomed by many "foreign" Apple owners.

As to Mr. King's complaints about the keyboard, it is surely a matter of horses for courses. Certainly all the comment I have heard in this distant land is totally in favor of the key placement on the Irish keyboard, as it is nearly identical to the keyboards fitted standard to typewriters available here. Mr. King also fails to mention that all such overseas Apple *IIe*'s, unlike their homegrown brothers, have a second character set fitted which is selectable by a switch under the front of the machine. This is of course a great selling point in Europe, and even in this country has created some excitement among teachers of foreign languages.

I am sorry to hear that Mr. King has found some incompatibilities with software written for US machines — foreign readers would no doubt be keen to hear details on these problems so that they could avoid or overcome them. Incompatibilities of US software which is otherwise ideal will not accept dates in the international format, nor four-figure postcodes, nor three-letter abbreviations for states. Software developers would do well to keep an eye on the other half of their potential market rather than effectively limiting their sales to the US.

A.M. Gainford
Robson House
The Shore School
North Sydney, NSJ, 2060
Australia

Dear Sir,

We have the following comments to make regarding Richard King's article in your magazine.

1. The UK Apple *IIe* was designed in the US and was not, contrary to Mr. King's claims, modified in any way by the Cork plant. The only significant electrical difference between the US and UK Apple *IIe* logic boards is the addition of PAL circuitry to allow UK units to function properly with PAL color monitors and, using a modulator, standard UHF TV sets.
2. Due to the extra space required by the PAL circuitry in the UK Apple *IIe*, the 80-column slot (auxillary slot) was moved to the center of the board in line with slot 3. Slot 3 was chosen because it shares the same I/O space as the auxillary slot. Therefore, if an 80-column card is installed in a US or UK Apple *IIe*, you cannot activate any peripheral card in slot 3 that requires I/O ROM space (i.e. a printer, Disk II card, etc). The decoding circuits in the US and UK logicboards are identical despite Mr. King's claims.
3. The only difference between the US and UK Apple *IIe* keyboards is that two little used keys have been moved, namely the "backslash" and the "proportional-to" keys. This is to accommodate a larger vertical return key in the UK keyboard. The larger UK return key is much easier to press with your little finger than the smaller horizontal one found on the US model.
4. Mr. King states that incompatibilities have been found when running US Apple *IIe* software on a UK machine. I don't know of any such software. I use a US Apple *IIe* at work (but with a 220 volt power supply!) and a UK machine

at home, yet I have never found any incompatibilities.

To summarize, the UK Apple *IIe* is as good as identical to its US counterpart with the exceptions and advantages of PAL color and two language sets (UK and US).

We are always willing to engage in constructive criticism of our products and indeed the development of our current and future product range is significantly impacted by customer inputs and reactions. We also recognize that we are constantly faced with economic/business decisions which may not always allow us to produce the "ultimate product" at a price which our customers can afford.

Michael Spring
Marketing Services Manager
Apple Computer (UK) Limited
Eastman Way, Hemel Hempstead,
Herts. HP27HQ
England

We are advised by Videx that this is not entirely true. A Videx UltraTerm in slot 3 of an Apple IIe will be selected by PR#3 and overrides the auxillary slot. (This also corrects the A.P.P.L.E. Doctor, Feb. 1984, page 62.)

Overseas Apples

Dear Val,

This letter is prompted by a somewhat misleading answer from the A.P.P.L.E. Doctor (Jan '84) to a reader's query regarding using computing equipment in foreign countries. Some time ago I purchased an NTSC color monitor and converted my Europlus Apple][to an American video output. With this experience, I was able to help a friend convert an American Apple to work here. The following factors would be important if an American were bringing a machine to Australia. Most of these issues would be equally relevant in Europe.

If you haven't decided which computer to buy, then Apple is a good choice for world wide user group support and reliability in climatic extremes. Your warranty is probably not portable. In a Kanute like effort to stem the tide of Asian look-alikes, it appears to me that local Apple dealers are proving somewhat belligerent to any Apple computer which does not have an Australian serial number.

It is not a good idea to use stepdown transformers. The small transformers which work at 60 Hz. use small laminations which are not adequate to disperse the extra heat generated when the same transformer is used at 50 Hz. If you must use transformers, purchase them here as they will be built for 50 Hz. operation. They will be large, heavy and very expensive. *You don't need a transformer.* It is easy to convert Apple power supplies from 110V to 220V. There is a black jumper lead at the switch end of a 110V power supply. The most time-consuming aspect of this conversion is breaking into the case!

Greater problems arise when considering peripheral equipment such as monitors and printers — generally it is a much better idea to purchase high quality equipment when you get here — you will find that it holds a good resale value when you leave to go home.

1) Modems must be purchased here to ensure a peaceful co-existence with telephone authorities which are semi-government bureaucracies.
2) Printers should be bought here — supplies of ribbons, daisy wheels or printheads for slightly different models of even popular brands can be difficult.
3) The multiple system TV sets recommended by the A.P.P.L.E. Doctor are very expensive and would be of little value when you got home. They would have little resale value here for the same reason.
4) Most of us discover that it is not a good idea to use the family color TV as a monitor. Buy a good quality PAL color TV for family use when you arrive — sell it when you leave. Buy a good quality monochrome monitor for your computer — if it is dual voltage you won't need to sell it.
5) If you really need color for your computer output, greater problems arise. Options include:
   a. Buy a NTSC monitor — if it's dual voltage, take it home with you.
   b. If you have an Apple][, buy a PAL card for slot 7. Local cards work better than Apple PAL cards. You will need to change the main crystal to have the correct line frequency, and you will need to resolder jumpers as per Reference Manual p.11.
   c. The designers of the foreign *IIe* were bright enough to include a convenient keyboard switch, but were sadly lacking when they put their inferior output stages on the mainboard. The old slot 7 was much more flexible. One solution for *IIe* owners is to buy one of the new 80 col. cards which include RGB output — if you can't get a dual voltage RGB monitor, at least it will have good resale value.

H.J. Hyde
14 Lovely Valley Rd.
FLAGSTAFF HILL 5159
South Australia

WORD-PROCESSING ?
apple][+, //e **ASSEMBLING ?** ➧ $75 p&p free
Here's real power at low cost !

WORD-PROCESSOR: ★ Simple to learn. Simple to use.
★ "Swopping" version for 48K Apple][Plus; "Enhanced" version for 64K][Plus and Apple //e; appropriate version automatically loaded at boot-up.
★ Compatible with full //e keyboard; makes use of all keys including arrows and "open-Apple".
★ No extra hardware needed, (16K card an advantage) yet you get upper and lower-case on screen.
★ 55 characters/line on screen (printing to any number).
★ Totally menu or prompt-line driven: your options always clearly displayed.
★ Key-letter commands (press F for Find, D for Delete).
★ Upper and lower case from keyboard, with temporary-upper-case and upper-case-lock; recognises standard shiftkey modification and enhanced keyboards like that on the Apple //e and Apple-copies.
★ Features a type-ahead buffer and allows insertion of characters like \ [{ | } and all control characters; printer control sequences are simple!
★ Full, easily-varied formatting of printed output includes page headings, indentation, autospacing, right- and center-justification, use of definitions.
★ Has Preview mode for viewing the final format.
★ Includes a Mailer system with selective mailing.
★ You can even insert pictures in your text!
★ Clear 85-page manual includes tutorial for Editor.

& ASSEMBLER: ★ Fully integrated with the complete word-processor.
★ Super fast (250 lines of source in just 9 secs!).
★ Assembly errors can be corrected immediately by returning to the offending line in the editor.
★ Uses standard Apple 6502 assembly language.
★ Assembler would be great value alone!

FULLTEXT::55
Purchase or details from:
Spacific Software
PO.Box 60381,Sunnyvale,Ca
94088-0381 (Ca res +6½%)

Circle 16 on Reader Service Card

A.P.P.L.E. Doctor

bruce f field

JERRY Battenhouse's letter in the November 83 Write -A.P.P.L.E. column has done wonders for my mail. For those of you who missed it, it was a continuation of the discussion started in this column about doing superscripts and subscripts with the C. Itoh 8510AP and Apple DMP printers. Jerry correctly pointed out that with the Apple printer, it is not possible to produce superscripts and subscripts by moving the printer platen up and down in the middle of a line. Whenever a linefeed command is sent, the printer also performs a carriage return. It is possible however to only move the platen using the C. Itoh printer.

First put the printer in the incremental print mode by sending "ESC [" to the printer, then forward and reverse linefeeds may be accomplished without a carriage return. This is one major difference between the C. Itoh and the Apple printer. This limitation rules out using most word processors to move the platen up and down for superscripts and subscripts. The Gutenberg word processor (and I assume the Gutenberg Jr.) can do superscripts and subscripts with either the Apple DMP or the C. Itoh 8510AP. I am not exactly sure how they work, as there are two ways of getting around the above limitation with the DMP.

It is possible to print all the superscripts first, then the main text line and then all the subscripts. This would require three passes for the line. Or, an alternate character set can be downloaded into the Apple printer (true also for the 8510 if it has the extra buffer). In this case small characters on a 5 by 5 matrix can be built to print at the top of the line or the bottom. They don't really extend beyond the regular text but because of their size are taken for superscripts or subscripts. This is the technique that Epson uses with their MX-80 and Graphtrax Plus option. There is one program I know of to download characters, FONT DOWNLOADER & EDITOR by Rak-Ware and it is advertised in most of the Apple magazines.

Question

This is mostly a question about preventive medicine since the darn thing has never failed in six months or so of use, but ???? Hex dumping an Applesoft program is very instructive. To make it easier, I wrote the following TEXT file:

```
PRINT "TURN ON THE PRINTER
   AND PRESS SOMETHING"
FOR X=1 TO 128 : X=PEEK
   (49152) : NEXT : POKE 49168,0
PR#1
PRINT
PRINT: POKE 60,PEEK(103) : POKE
   61,PEEK(104) : POKE 62,PEEK
   (175) : POKE 63,PEEK(176) :
   CALL -589
PRINT
PR#0
```

All that needs to be done is LOAD programname and EXEC APHID ("APHID" is the name I saved the TEXT file under — would you believe "Applesoft Program Hex Instructional Dump"?)

As I have said, it has worked 100% for me for many months. But, I read recently in What's Where in the Apple *that the Y-register has to be initialized to 0 before the CALL to address -589. (This would be easy to do: insert the three commands CALL -182 : POKE 71,0 : CALL -193 in front of the CALL -589.) Is this really necessary in light of my success? Why, or why not? Is it possible that the Y-register will automatically have the right value in this context? If so, how can I (primarily an Applesoft programmer) know what is safe and what is not?*

Answer

Let's start at the beginning and see what happens. The routine you call at -589 is a monitor routine that takes the starting and ending addresses from locations 60,61 and 63,64 which you have set to the beginning and end of your Applesoft program. It is not true the Y-register must be set to zero for this to work properly, the first thing the routine does is to set both the X and Y registers to their proper values. It appears the What's Where in the Apple is incorrect on this point.

It is not generally reasonable to assume that any of the registers will be set to the same thing every time you call a machine language routine, what they contain will depend on the previous sequence of instructions. If we assume that the Y-register does indeed have to be set to something specific, your CALL -182 etc. sequence will not work. The problem is that between the CALL -193 and the CALL -589 that follows, Applesoft regains control and immediately changes the Y-register (also the A and X-registers). If you wish to set a 6502 register there is no simple universal call to do this from Applesoft. One way around this is to write a short machine language program that sets the register and then jumps to the machine language routine you really want to run. Then you can call your new routine from Applesoft.

As for your last question, the only way you can be sure that the registers are set properly is to explicitly do so or learn how to read assembly language programs (the former is less painful).

Question

I am using Applewriter II to print a manuscript but no matter how I set the top and bottom margins the first page always comes out with the wrong number of lines. Is there a bug in Applewriter?

Answer

The problem is undoubtedly that you are changing the top and/or bottom margins from within Applewriter using the (.) dot commands. By the time you execute the .tm (top margin) command, it is already too late to do anything about the top margin on the first page. Applewriter uses the default print values in the Print Command Menu until dot commands are encountered in the text. To make the first page come out correctly, set the default print values to the same values that the dot commands specify. This procedure is explained in the Print section of the Applewriter manual.

This problem also shows up if a header is used and the left margin is changed at the beginning of the file. The header on the first page is printed according to the default print values and may not line up properly with the remaining text. The header continues using the value for the left margin that was set for the first page, so changing the left margin on succeeding pages will not affect the header.

Question

I have the //e with the extended 80-column card and WordStar version 3.01P.

I'd like to use the auxiliary memory as a pseudo-RAMcard in which to stash one or more of the WordStar COM and OVR files, thereby freeing disk space and hopefully WordStar's rather geriatric performance on the Apple. How do I access the additional memory under CP/M, and what patches to WordStar and changes to CP/M are necessary to give RAMcard-like results?

Answer

(What I know about CP/M will fit in a nutshell so Murray Arnow was kind enough to answer this question.)

This requires an involved reply. I will briefly state that there is no simple way to alter the BIOS to use the extended 80-column card memory to become a pseudo-disk. I seem to recall that there may be commercial packages available to perform the required patching, but I am unable to give any recommendations. There is a simple way to speed up WordStar. Create a WordStar disk by first formatting a new disk and then PIPing over the WordStar files in the following sequence: first the WordStar overlay files and then the WS.COM file. The WordStar files should appear first in the disk directory. If the latter is done, then disk access time can be significantly shortened.

Question

I would like to change the file name used for disk boot up without initializing a new disk. Is there some way to do this?

Answer

As you know, when a disk is initialized a greeting program (usually named HELLO, although, it doesn't have to be) is saved on the disk. The easiest way to boot up with a different greeting program is to rename or delete the original and save the new program under the same name as the original. If for some reason you actually want to change the name, you can do this if you have a disk zap program. This is a program that allows you to read any individual sector on the disk, modify it, and write it back to disk. The greeting program name is stored on track 1, sector 9, starting at the $75th location. Read this sector into memory, change the name to what you want, and write it back to disk. It's a good idea to make a backup before doing this, as it is fairly easy to make a disastrous mistake and permanently wipe out some useful information.

Question

*When drawing on the high-res screen with HCOLOR=5 the X-coordinates with X=13+14*I (0<=I<=19, e.g. 13, 27, 42, 55, . . .; I will call these peculiar) seem to be somehow unique. If I plot a vertical line at a peculiar position and another one at a normal position, the first one is darker than the second one. I am able to observe this on my monochrome and RGB equipment.*

Try:

```
HGR : HCOLOR=5
HPLOT 13,0 TO 13,100
HPLOT 43,0 TO 43,100
```

If I plot a third line to the left of the peculiar line, nothing strange will happen. But a third line just to the right (HPLOT 15,0 to 15,100) will make the peculiar line brighten up.

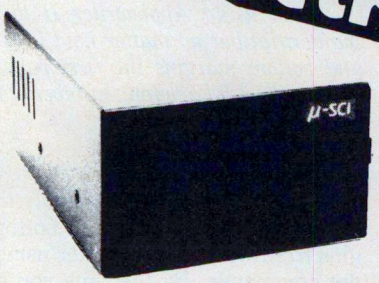
Do you know, whether the above can be seen on all Apple][s, or is there something wrong with my Apple?

Answer

There is nothing wrong with your Apple, this is "normal" for all Apple][s and //es. Your peculiar lines result whenever you plot in the last dot position of an even byte (this is what you call a peculiar line) and the next byte to the right has the color-shift-bit set opposite of the color-shift-bit in the even byte. Each byte in the graphics memory has 8 bits, 7 of which are displayed to the screen, and the 8th is the color-shift-bit. If the color-shift-bit is off (HCOLORs 0 through 3) then the next 7 dots can be used to produce either green, violet, or white. If the color-shift-bit is turned on (HCOLORs 4 through 7) then the next 7 dots can be used to produce either orange, blue, or white. If the even byte has the color-shift-bit on and the next byte has the color-shift-bit off, the line is fainter than normal. When you plot a line in the next byte that turns the color-shift-bit on, even HCOLOR=4 which plots a black line, the peculiar line returns to normal. Conversely, if the even byte has the color-shift-bit off and the next byte has the color-shift-bit on, the peculiar line will be brighter than normal. I can't explain exactly why this happens, but it appears to be a bug in the Apple graphics hardware.

An excellent article, Color Me Apple by M. A. Iannce, appeared in the November 1982 issue of *Call -A.P.P.L.E.* that explains most of the Hires graphics idiosyncrasies in layman's terms. I recommend this article to those of you who want to know how your Apple really generates colors.

Laing Electronics



μ-SCI
MICRO-SCI
A2 DRIVE

$229

BUY THE FIRST AND THE BEST APPLE COMPATIBLE DRIVE!

A2 DRIVE WITH CONTROLLER $309
(FREE SHIPPING IN USA OR CANADA. A2 MODELS ONLY.)

A40 (40 TRACK DRIVE W/O CONTROLLER) $284
A70 (70 TRACK DRIVE W/O CONTROLLER) $365
— CONTROLLER FOR A40 OR A70 DRIVES $ 79
A3 APPLE III DIRECT REPLACEMENT $299

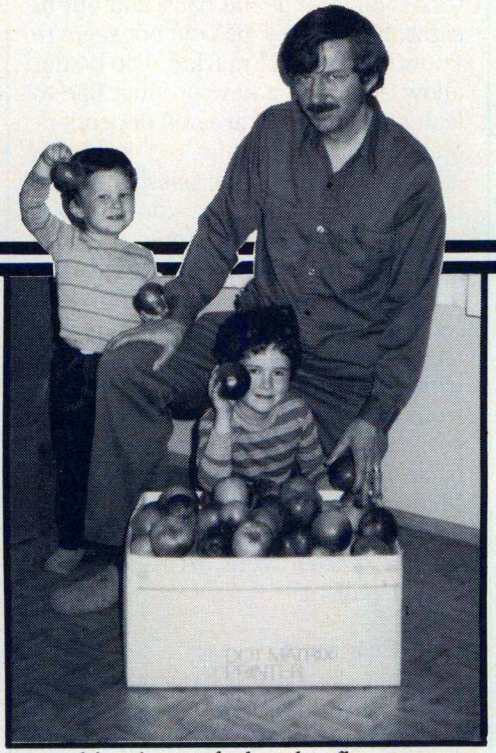
10382 Stanford Ave.,E-203, Garden Grove, Cal., 92640
Phone (714) 534-4216

Check or Money Order. — NO CREDIT CARDS! Check's Allow 10 BANKING DAYS FOR CLEARANCE. Prices and specifications subject to change without notice. Add $7.50 for UPS surface. Cal. Res. add 6% Sales Tax.

Circle 18 on Reader Service Card

The Northern Spy

rick sutcliffe



YOU can tell a lot about an industry when some of its retailers get together. The Fourth Annual Pacific Coast Computer Fair (held September 17 and 18 in Robson Square, Vancouver, BC) might not have the drawing power of Las Vegas or San Francisco, but it does provide a local window onto the fastest growing and fastest changing industry the world has ever seen.

This space will not provide just a look around the rooms however, but a sampling of what is coming in the years ahead, hung on the framework of the fair (or on that of this humble scribe's opinion).

Perhaps Nathan (my five year-old) thought that there would be rides and cotton candy. At any rate, he was less than altogether happy with me as we wandered among the exhibits. Having him carry the bag into which we put the freebies helped, but the fair he was interested in was the one upstairs. It isn't too interesting when all you can see are knees or the underside of tables and all you can think about is food. There was more than that for me however, and this provided a good place to begin some observations of the Vancouver (and larger) marketplace.

Amber

Remember that word. It will soon describe all the video monitors sold, as it generally does now in Europe. Forget all that nonsense you sometimes hear from the self-appointed saviors of the poor benighted CRT typists who claim on their behalf all sorts of medical problems and assign them to radiation exposure.

The real problem has always been tension. This is brought on by a combination of poor posture, sitting too long at a stretch, high pitched CRT noises, and eyestrain caused by the combination of harsh fluorescent lighting and flickering displays.

Changing to a high persistence amber monitor and softening the ambient lighting would go a long way toward solving the problem. Frequent short exercise breaks and better seating conditions would solve most of the rest. If there is previous ear damage and CRT exposure causes persistent ringing, it may be necessary to place a sonic baffle around the system, though the computer itself is rarely the culprit. Radiation indeed!

There were a number of these amber monitors at the fair, some as a part of total packages, and others as add-ons. In the latter category, I liked the Amdex Video 300A and have now bought one. It is unquestionably the best monitor I have yet used. In the former category were a number of commonly shown systems, including the now orphaned Osborne Executive. Next year, everybody will be using them.

Down in the DMP's

Another key peripheral much in evidence and which I have also recently obtained for Trinity is the Apple dot matrix printer (C-Itoh with Apple's enhancements). The DMP is state of the art in this department. It is extraordinarily versatile and can do as much or more than any other high quality dot matrix printer. (Some of these cost considerably more.)

For hackers, it has user-definable character sets; for educators, it appears durable enough to take very heavy use (alas, EPSON is not, I have discovered); for verbose programmers; it is fast and bi-directional.

Hardware and/or software switches allow the selection of seven different character pitches in seven different languages. Double strike and elongated modes are also available as are software control of margins, spacing, bit image graphics (High-Res), uni/bidirectional modes, tabulation (horizontal and vertical), and line feed pitch to 1/144 inch. The optional 2K RAM will serve either as buffer or storage for a downloaded character set. This is the printer which forms part of the Lisa package, and the first spin-off from that to become generally available.

Any fly in the ointment? One. For some bizarre and inexplicable reason, there is no manual included with the printer. For an extra $30 or so, you buy an accessories package containing this together with the warranty card (!). I assume that the collective outrage over this new and ridiculous policy will soon cause it to be revoked. I hope so.

That's Nathan(5) and Joel(3) with me in the picture, by the way with our crop of Northern Spys overflowing the DMP box. It does make a great apple crate. By the way, the DMP has now been replaced by the Imagewriter—very similar, but with a serial interface . . . Yet another new wave?

Time, Gentlemen

There were several time cards in evidence; the one I recently bought for Trinity is from Applied Engineer-

ing, P.O. Box 470301, Dallas, Texas 75247. It is plus/e and Pro-DOS compatible and displays the time in four modes including Apple Clock format. It comes complete with a disk full of utilities (two sides — BASIC and Pascal) to make use of the device, and at $129 seems like a bargain.

Other Peripherals

Much of the Apple's juice has always flowed through those eight twigs sprouting at the back of the main board, of course. Naturally, all sorts of plug-in peripherals were in evidence, from S100 boards to PC add-ons to a healthy selection for our machines.

One of the most exciting of these is the fabulous new Ultraterm from Videx. For a wordsmith like myself, having a 48 row by 80 column or 24 row by 132 column (or other) modes available at the touch of a key is a real experience. Eight display modes, crisp and clear beyond belief . . . Prediction: every Word Processor and Spreadsheet on the market will be compatible within four months. Apple may just build the thing in — it is the single most important display peripheral ever made for a small computer. Buy one, I did.

Editor's note: a review of the Videx Ultra-term appeared in the March issue.

The next most significant is additional memory, of course. The best thing on chips right now is the Applied Engineering 128K extended 80-column board for the *II*e. Titan (Saturn) is in there too, with a similar (and much more expensive) 192K version. Macrotech also makes some nice memory boards and associated software — great to see Paul Bowler's North Vancouver firm there with a good display and obviously doing well . . . catch his ads in *Byte, Call -A.P.P.L.E.* and other publications.

Portables

There were lots of these, and there will be more as the number of chips in computers decrease and disk drive manufacturers continue to shrink their packages. The Hyperion from Dynalogic was a good example. Shown by several stores, it has nice clean lines and seems to have had the benefit of some professional styling advice.

That could not be said of the Osborne's Executive. Though the pioneer in this part of the industry, they had never realized that connectors should be at the back and out of sight. Sadly, they also did not seem to know that in a hot market, you do not allow details of a new product line to leak out when you are still dependent on sales of the old one.

We will mourn the passing of Osborne to the bankruptcy courts, but it of course represents only one of many casualties. The maturing microcomputer market has now entered a shakeout phase. Some big name manufacturers will be gone from the computer business a year from now — like Osborne, with the corporate ship sinking altogether, or back whence they came . . . making filing cabinets, chips for someone else's machines, or arcade games. (These words were written before the TI pullout.)

New Entries (1)

That prospect has not given much pause to a flood of new entries to the market however. Two possible survivors which were shown were the Qume QVT102 and the long-awaited Sony. Both offer some fresh concepts, and both come from companies which have reputations for producing reliable, quality products. The modular Sony is an especially nice idea, and if it has adequate software backing, it should catch on with that company's traditional customers.

New Entries (2)

Despite some recent demises also at the retail level, there continue to be those who think they can smell a dollar in the selling of computers. They probably can if they have both the requisite business acumen and technological literacy to carry it off. Mind you, there are a few stores in the local arena now which obviously have neither. Let us hope they will soon vanish from the scene.

This show saw entries by Kelly's Stereoland (showing a wide range of NEC products) and London Drugs Computer Department (lots of software). Well, they make it work selling eyeglasses, do they not?

One hardly counts the department store efforts so far — these two represent the first wavelet of mass merchandisers into what has been somewhat of a closed specialty shop in our particular area. I fully expect to see more of this sort of thing, for the computers of the future will be *user* devices, rather than the exclusive property of hobbyists. They will be sold right beside televisions, stereos, sewing machines and refrigerators, and be featured weekly in the pulpy mass of flyers you scrape from your rain-soaked porch.

Business oriented systems will continue to be sold by specialists. Many more manufacturers will market direct as does IBM, and these changes will mean that the present retail stores will have to hustle to remain a factor in the market. Some will, some won't.

All in One

Actually, there are people trying to take the bundled package much farther than Osborne and its imitators did. The Access has a built in printer and amber monitor (there's that word again) at $2500 and there will soon be numerous packages of this sort to choose from.

This particular wave will crest somewhat lower though. For truly lightweight and truly portable use, a mechanical printer simply adds too much weight unless a lot of quality is sacrificed. Now if dry copying principles can be applied to the production of an entire sheet of paper at one time with no moving print head, we might get somewhere.

On second thought, that implies major advances in plain paper copying, doesn't it? Would it be impossible to combine a SCD (Sheet Copy Device) with an 132 column by 66 row flat display screen, a billion bytes of storage on line (in an optical disk?), a telephone interface, two or three million bytes of main memory and a full-sized keyboard auxiliary to the voice input/output? Would it be impossible to cram that into a 10kg briefcase package? Well would it? . . . No.

Would it be possible someday to eliminate the keyboard and have voice I/O only, put all that storage into main memory and then slip the entire thing into your shirt pocket? Would it be possible by 1990? . . . Yes.

Some Other Mechanicals

I wouldn't like to disparage what is now being done with mechanical printers. Brother's new little printer was there, but was a little pricy I

thought. The TeleVideo booth had something more to my liking — a compact Olympa electronic typewriter chugging along as a printer at 14 characters per second.

Dual purpose typewriter/printers are ideal for the small office or home where the workstation cranks out only a few pages of "letter-quality" printout. They would be less suitable for production runs of form letters, but you can still use the typewriter function — a big plus. More of this, please.

The Genetic Engineers Again

Naturally, there was the usual selection of "IBM compatibles" — which never really are and lots of Apple "clones" — which aren't either.

In the case of IBM, proprietary designs prevent 100% compatibility — even where the same accessories will work perfectly, software may not. In the case of Apple, most of the hardware pirates change the physical design enough to avoid patent infringement but steal outright the more than 16000 bytes of code in the built-in memory — right down to the programmers initials.

It is always interesting to discuss this issue with the people involved. The owner/decision maker invariably claims something like "Apple's rip-off pricing structure justifies our making (or selling) their machine." A conversation with a retail employee usually goes like this:

"Is your Mango II Apple compatible?" "It certainly is. It will run all Apple software."

"Does it have the Applesoft language built into its memory then?" "A modification of Applesoft, yes."

"How much has it been changed?" "Well, I don't know, but it is different. That's how we avoid copyright problems."

It's at that point when I usually swallow very hard and leave. The fact of the matter is that I have yet to test an Apple "workalike" with even 2% of the copyrighted code altered in any way. (The average is thirty bytes.) Well, Apple has won their appeal against Franklin in the States, so it should be "good-bye clones" down there at least. Their end cannot come too soon.

For both Apple and IBM there is a solid, almost invincible reputation for quality, reliability and customer care. Their imitators have none of these things, and the wise customer knows it and spends money accordingly. The type of total confidence that has been given to these two companies is incredible, but it is well earned. Others may also achieve this, but it will take time. I would say that only Heath/Zenith is close.

Lisa

Yes, and she was there — such a lovely lady. For your personal desktop "gal Friday" there is nothing like her. Oh, and the price of her hire just dropped. She'll come and stay at your place (sans bundled software) for under ten grand (CDN) now. Now with a price of (US) $3495 (a second reduction) and a Mac-compatible 3½" micro-floppy, Lisa makes a very attractive offering. If you're in that league and want the very best, buy. It will be some time before any realistic competition emerges.

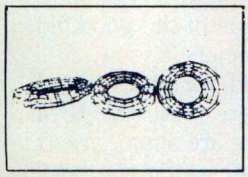
Now for our College system — I'd like a Lisa running Unix (now available) and a network of computers/terminals (MACs?) with at least 200 megabytes of disk storage. Any big donors out there ready to make a commitment to a private sector University? It is to dream.

The Mac

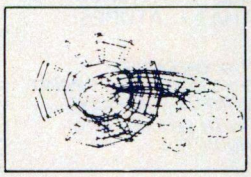
Speaking of what the big Apple is up to (what else do we do around here?), the new Mac should be available by the time you read this. Featuring easy transportability, very advanced bundled software, and the promise implied by the 68000 chip that Lisa technology can be shared — this newest member of the family should rewrite the book on small computer applications. Since it will not have the old Apples' hardware expandibility, it may not be as warmly received by hobbyists as it will by the mass market, but say hey — at these prices a lot of people will afford

Don't Settle For One Point of View

Step into the world of 3D



Create, edit, move, rotate, animate ... produce amazing graphics effects.



It's easy - with Metalogic. A comprehensive interface to SubLogic's powerful graphics package. But forget the hex, PEEKs and POKEs. Just combine Metalogic's "&" commands with regular BASIC statements and you've mastered the 3rd dimension.

Why settle? Let your possibilities be limited only by your imagination — not your software.

Req. SubLogic A2-3D2; Apple][Plus & RAM Card (or Apple //e); DOS 3.3
$75 (or $180 including complete SubLogic graphics package)

Apple][Plus, Apple //e, DOS 3.3 are trademarks of Apple Computer Corp. A2-3D2 is a trademark of SubLogic Corp. Metalogic is a trademark of Artemis Systems, Inc.

ARTEMIS SYSTEMS, INC.
41 Parkview Drive, Millburn, N.J. 07041
(201) 564-9333

MasterCard VISA

New Jersey res. add 6%
3rd class postage pd all orders
add $2 for 1st class mail.

All software unprotected

Also from Artemis

Telephile $20
A database to keep tabs on all your important contacts-friends, relatives, business associates
- Simple to use
- Complete screen editing
- Flexible search includes wildcards
- Hard copy in variable formats
- Approx. 200 116-character records in each file
- Includes Name, Address, Phone, & Comments Fields

Pascat $25
Catalog all your Pascal disks for easy reference
- Automatically reads and sorts all files
- Creates master catalog and saves it
- Screen list or hardcopy
- Flexible search capabilities including wildcards
- Requires Pascal

Screen Unit $45
Terminal I/O unit a must for your Pascal library
- Device independent
- Field oriented with optional defaults
- Built-in editor (insert, replace, delete, clear, etc.)
- Requires Pascal
- Source & object code included
- Input checking prevents user error
- No licensing fee for commercial use

Circle 19 on Reader Service Card

both, especially at selected universities in the College Consortium where the prices to students and faculty will be severely discounted.

Castle Gravenstein

Oh, no, I don't know what he will call the thing, but when the Woz finally gets around to a *real* upgrade of the *II*e, we should see some more sparks fly. My shopping list includes the 65816 chip with its 6502 compatibility, 16 bit data handling and 24 bit address bus, a better display (Ultraterm style), a more convenient memory handler, keypad included (finally), clock, alternate processors and better disk drives.

We will not have to wait a year for the //f (or whatever) and it will still be compatible to the older line (as well as to the Mac?) Say September? In the meantime, let's see some software for the slightly stripped down 65802 chip which has the (flag selectible) wider data capacity and the extra op codes, but lacks the pins to address the 16M of memory. Plug it in to the old Apple and run with it, it is completely compatible with the 6502.

Peanuts and Popcorn

Now they weren't there of course, because even the PC Jr. was introduced only between drafts of this piece (these things can take months to germinate), but I am truly astonished by what IBM is trying to pull. If anybody else tried to foist a dud like this on the market, they would be broke in four months. It lends credence to my claim that the PC has done as well as it has only because of a combination of luck and the name on the plate. Apple Computer Inc. executives must be laughing up their collective sleeves, now that it is so obvious how much better the Mac is. No contest.

Lies and More Lies

If you read be-Twain the lines, you will know that I refer to statistics. Several articles recently have purported to quote sales figures showing that IBM will catch up (has caught up) to Apple in the latter part of 1983. Nonsense. If their figures for big mother blue are right, then mine say that Apple will have a comfortable lead far into 1984, and perhaps well beyond. Would you believe a million Macs alone in that year? Would you believe almost as many *II*e (//f) machines in addition?

Books

Computer Supply had them — lots of them. A few others carried much smaller displays, but this is an information business is it not? Many more magazines, books, technical brochures (some ran out), software packages, and miscellaneous small paraphernalia could find a ready market here.

My problem with the book publishers is the very high prices. Many computer people act as though they had limitless money for accessories. The book trade knows this, and generally prices our materials at double or more the price of comparable ones in other fields. Another factor driving such prices up is the fragmented market. After all, there are only just so many people who want a 500 page tome on interfacing the 790000 super MPU chip to a 7922 VIA.

The Clubs

Hosts and sponsors of this event, the user groups were out in force. Not unexpectedly, the Apple group took up the largest single space, but I spotted two clones on their table (boo, hiss). The TRS groups taken together were the most numerous for any one brand, even though there was no Radio Shack equipment being shown.

These self-help groups have been invaluable in the growth of the microcomputer industry. I daresay that without them, IBM would not have had a market worth stepping into. They have fostered the kind of spirit of cooperation and genuine non-profit concern for others upon which our society was once built. Let's hear it for some of the good old ways in our new society.

But as I headed for the door, I wondered . . . where were the IBM users groups? There are some, aren't there? Or are businessmen too busy to help each other? Hmm . . .

Exhausted

Well, back upstairs in the Pacific Centre food fair, Nathan finally got his root beer. I had my story. Mission accomplished. The last and most difficult task was to navigate the streets of the wicked city back home to Aldergrove.

Next month, we'll do a tiptoe through the industry again, but this time looking at software trends.

~~TRIPLE!~~
~~DOUBLE~~ DENSITY. . . ON STANDARD APPLE DRIVES?

Introducing TRIPLE PLAY

A COMPLETELY . . . . . SELF CONTAINED ENHANCEMENT OF APPLESOFT THAT FEATURES:

* NUMERIC DATA COMPRESSION STORE 3½ TIMES MORE INTEGERS! STORES 3 TIMES MORE REALS!
* SYNTAX CHECKING WORKS EXACTLY LIKE **PRINT** AND **INPUT** STATEMENTS!
* TEXT AND NUMBERS CAN BE INTERMIXED!
* USE WITH SEQUENTIAL OR RANDOM FILES!

USE TECHNIQUE WITH ANY DOS OR DISK SYSTEM (FLOPPY, HARD OR RAM)

AND IT'S FASTER THAN NORMAL I/O ROUTINES!

IF YOU STORE NUMBERS THEN YOU NEED TRIPLE PLAY. GET AT LEAST 3 TIMES MORE NUMBERS PER DISKETTE.

UNPROTECTED PROGRAM DISK WITH COMPLETE DOCUMENTATION ONLY . . . $19.95

TO ORDER: Ask for "TRIPLE PLAY"

send check or money order to:

GRAYSOFT
P.O. Box 515
Buffalo, WY 82834

307-684-7257

PLEASE ALLOW 2 WEEKS FOR DELIVERY. FOREIGN ORDERS MAY REQUIRE MORE POSTAGE.

(NOTE: APPLE & APPLESOFT ARE REGISTERED TRADEMARKS OF APPLE COMPUTERS INC.)

Circle 20 on Reader Service Card

Koala Pictures

lorana a jinkerson

AFTER reading about the Koalapad in the September issue of *Turtle News* and seeing a demonstration using a joystick (not the pad), I became totally convinced of my desperate need to acquire the critter. As a result of this uncontrollable desire, I scribbled a note off to Santa and lo and behold the little old man came through with the package for me.

I have found the Koalapad easy to use, interesting, and a useful tool for working with graphics on the computer. At the present time, my Koalapad is used mainly by young children (5 on up) as a drawing or scribble sheet. The children easily learn how to control the cursor and get the shapes and/or designs they want. Even more so than with the Logo turtle, they derive a lot of pleasure from creating pictures on the computer screen with such ease.

We have, however, experienced one problem for which I have devised a solution. Let me begin by describing the problem. Children I'm in contact with have enjoyed creating pictures with the computer, Logo, and now with the Koalapad. They do not, however, like to see their picture disappear forever when they turn off the computer. Solution — SAVE the picture in Logo using SAVEPICT, or when using the Koalapad, use its SAVE routine. At least then the child can LOAD the picture back into the computer's memory and show daddy, grandmother, or his friends the masterpiece he created. Great, except that at 33–34 sectors a picture, a disk soon becomes full — often with pictures not really worth saving except to please the child.

Up until now, I used Logo with a printer. I had them READ in my hardcopy procedure first, then when they completed their drawing or design, they instantly printed it. In fact, we usually made two or more copies so that I could have one to hang on our wall in the computer room and they could have at least one to take home to show others their work. They really delight in having their pictures to carry around or post, and I seldom have to SAVE a picture unless we plan to work in it again or want to use it in another program.

With the Koalapad, there is not a procedure to print out the picture — only to SAVE and LOAD them. In order for them to print their pictures, they first must SAVE it with the Koalapad's routine, end the Koala program, start up a dump program which then LOADs the picture back on the screen, and finally get their copy of the picture. This becomes quite a chore for young children.

The Final Solution

Insert this PRINT routine directly into the "BY STEVEN DOMPIER" program on the Koala disk:

```
675  PRINT: PRINT "P) PRINT
     STANDARD DOS 3.3
     PICTURE"
1005 IF A$ = "P" THEN 3000
```

Next, insert the subroutine dump procedure for your specific equipment at line 3000. I have a Grappler Plus and a Gemini 10X printer, so the following subprocedure accomplishes the task quite nicely. You need to adjust the control codes for your hardware set-up. Here is how mine looks:

```
3000 I$ = CHR(9)
3010 PRINT D$;"PR#1"
3020 PRINT I$;"G"
3030 PRINT D$;"PR#0"
3040 GOTO 20
```

It is possible to add options to the print routine as below:

```
3002 HOME: PRINT "ENTER
     OPTIONS (I,R,N)"
3004 INPUT Z$
3006 IF Z$ = "N" THEN Z$ = ""
```

You would then need to change line 3020 as follows:

```
3020 PRINT I$;"G";Z$
```

Again it would be necessary to use the codes appropriate for your equipment. In this case, "I" prints the picture in inverse, "R" rotates the picture 90 degrees on the paper, and "N" prints the picture normally.

At the same time, you can also add an END choice to the storage menu so you can leave the Koala program ready to go on to other adventures. Insert the following to accomplish this:

```
676  PRINT: PRINT "E) END"
1006 IF A$ = "E" THEN 6000
6000 END
```

With these simple additions to the Koala program any young child can have an instant copy of his work of art.

FLYING HIGH. . .WITH E-Z PILOT

ZOOM! Big Time Program

earl keyser

WELCOME back to Flying High, the column which will teach you the educator, or you, the parent, or you, the student how to write fun and neat lessons/programs with E–Z PILOT.

This time, we'll cover using labels in conjunction with U statements. And, we'll begin to do the things the manual *didn't* completely reveal.

First, let's talk about U.

U, which stands for use, is like a worker on an assembly line. It does one special thing over and over and over, again. It serves one special purpose and, when called on, does its one thing.

A really simple example would be this program which when run, prints:

```
1 R:Astronomy lesson
2 U:*T
3 U:*T
4 T:Little star.
5 E:
6 *T:
7 T:Twinkle,
8 E:
```

Twinkle,
Twinkle,
Little star.

The Use lines (2 and 3) tell E–Z PILOT to jump to the label following the colon. In this case, that is *T. Then, the next statements are executed UNTIL E–Z PILOT sees an E: statement. Now, watch this. At no time do my hands leave the ends of my arms.

When a U statement is active and E–Z PILOT sees an E: statement, the program *does not end.* Instead, it jumps *back* to where it came from. Actually, it returns to the line following the U: statement.

So, our example will USE *T to print "Twinkle," twice. Each time, it will jump to *T and, after the E:, it will jump back. Finally, it will print "Little star." and it will end.

Interesting, but not really useful, you say.

True. It was just a flash in the pan.

But this next program is a real, live program which my daughter, Amy (about whom you will hear much more) uses to work on her spelling lessons. Its purpose is to print each of 17 spelling words on the screen, erase the word and ask her to type it in. If my Mockingboard™ is hooked up, it will also speak. The speech adds a lot to the program.

Look through it and see if you understand it. A little explanation will follow. I have only numbered the first 34 lines of the lesson. AUTHOR, of course, would number them all.

```
1 R:Talking Spelling Test
     11/9/83
2 J:*S
3 *Y:
4 T:RIGHT!
5 S:50,100
6 E:
7 R:This subroutine asks for
    the word.
8 *W:
9 GW:
10 T@:5,1
11 T:What is the word
12 A:
13 E:
14 *S:
15 R:BEGINNING OF LESSON
16 T:(Ctrl and G)What is
     your name
17 A:$NAME
18 T:Hello, $NAME
19 T:
20 T:This is
21 T:a spelling test
22 T:
23 T:Can you spell
24 T:each word?
25 T:
26 W:1
27 *1:
28 GW:
29 T@:5,1
30 T:PRINCESS
31 U:*W
32 M:PRINCESS
33 UY:*Y
34 JN:*1
*2:
GW:
T@:5,1
T:CRYSTAL
U:*W
M:CRYSTAL
UY:*Y
JN:*2
*3:
GW:
T@:5,1
T:HANGING
U:*W
M:HANGING
UY:*Y
JN:*3
*4:
GW:
T@:5,1
T:STUFF
U:*W
M:STUFF
UY:*Y
JN:*4
*5:
GW:
T@:5,1
T:OUR:
U:*W
M:OUR:
UY:*Y
JN:*5
*6:
GW:
T@:5,1
T:LEARN
U:*W
M:LEARN
UY:*Y
JN:*6
*7:
GW:
T@:5,1
T:MEADOW
U:*W
M:MEADOW
UY:*Y
JN:*7
*8:
GW:
T@:5,1
T:KNEW
U:*W
M:KNEW
UY:*Y
JN:*8
*9:
GW:
T@:5,1
T:BEHIND
U:*W
M:BEHIND
UY:*Y
JN:*9
*A:
GW:
T@:5,1
T:TRACTOR
U:*W
M:TRACTOR
UY:*Y
JN:*A
*B:
GW:
T@:5,1
T:ROCK
U:*W
M:ROCK
UY:*Y
JN:*B
*C:
GW:
T@:5,1
T:ENOUGH
U:*W
M:ENOUGH
UY:*Y
JN:*C
*D:
GW:
T@:5,1
T:ROBINSON
U:*W
M:ROBINSON
UY:*Y
JN:*D
*E:
GW:
T@:5,1
T:THERE
U:*W
M:THERE
UY:*Y
JN:*E
*F:
GW:
T@:5,1
T:BUILT
U:*W
M:BUILT
UY:*Y
JN:*F
*G:
GW:
T@:5,1
T:SCARED
U:*W
```

```
M:SCARED
UY:*Y
JN:*G
*H:
GW:
T@:5,1
T:HAPPENED
U:*W
M:HAPPENED
UY:*Y
JN:*H
R:END
E:
```

Did you read it all through? If not, go back and do it!

Now, for the explanation.

Line 2 jumps over the two USE routines *Y and *W. Otherwise, we would fall into them and the program would end after printing RIGHT! and one little whimper. No bang, just a whimper.

*Y is the beginning of the routine we will USE if the student gets the answer correct. It will print "RIGHT!" on the screen in big letters, play one note and end by returning to wherever it came from.

*W is the USE routine which asks for each word. It erases the screen (GW:), sets the cursor down 5 and over 1, asks for the word and accepts an answer. Then, it returns whence it came.

Line 15 begins the lesson which asks for the student's name and stores it under $NAME. Next, we tell the student what to do.

Line 27 begins the actual drill. There are seventeen sections of this and some can be removed or simply not entered if you wish.

Why seventeen? Because an E-Z PILOT lesson can only contain 20 labels, and 17 words consumed them all. 17 not equal to 20? Ah, remember *W and *Y and the other one. Where? Remember line 14?

Anyway, line 27 is the first of 17 labels for lessons. In each section, the screen is erased, the cursor is set and the word typed. Then, because we will need it *every time,* we USE routine *W, which again erases the screen and gets the student's answer. Then, we return to our mainline program.

Next is the match statement to see whether they got it right. If so, the UY:statement calls *Y which rewards them. If they missed it, line 34 (and equivalent lines in other sections) returns the student to see the word again. Failure is not rewarded with a raspberry. Only success is rewarded.

And so on for sixteen more words. Thus, the asking and rewarding section in Talking Spelling Test in each section requires two lines of E-Z PILOT code. Without the subroutine, six lines would have been required. This would have meant an addition of 102 lines to the lesson!

My observation of Amy using this lesson is that short words, 3-5 letters, were gotten on the first try. Longer words took longer. But, she never took more than four tries for any word. Naturally, a 7-year old computer genius who speaks several languages and is working on her doctorate in astro-physics is not typical. Still...

Can it be jazzed up? Sure. I could add a picture after each word (but 17 pictures won't fit on the disk) or colored letters in the reward or a whole song or all of the above. I think that a little positive reinforcement is good. Too much, however, tends to make her forget what she is doing — working on her spelling words.

Enough! USE is the toughest concept in E-PILOT. Next time, more about S: and G:. Maybe a mention of the Koala pad. Maybe not.

Oh, try this somewhere in the above program:

```
C:LN=50
```

What happened? Why?
Until next time.

A Call A.P.P.L.E. Quickie

BECAUSE of GPLE's fast load process, it cannot be modified in memory and BSAVED. However, like many users, we do not care for a flashing cursor; so we have modified GPLE *on the disk* using a disk edit utility.

Using a "search" program, look for and locate the four-byte hex sequence of 29 1F 29 7F. It will occur in three sectors, once for each version of GPLE.

Change the 7F to 3F and write it to disk. The cursor will no longer flash. This patch has no effect on an Apple *IIe*. Please make sure you try this on a *back-up* copy.

Flying Low: A Bug!!!

If your E-Z PILOT is below serial number 200 **and** *you have an Apple II or II+* **and** *you have the shift key mod* **and** *you use it, you will never get capital P's. Only @'s.*

Why? Because the stupid author made a stupid mistake. If you have this problem, here is the fix.

Boot your Apple from the System Master then insert your E.Z. Pilot diskette in the drive.

LOAD AUTHOR.HG

Delete line 11 by typing:

11

Replace lines 10 and 14 as follows:

```
10 IF A$ = "@" AND LI$ < > "T"
   THEN A$ = "P"
14 IF PEEK (ST + 83) > 127 AND
   A$ > "@" AND ( LEN (LI$) > 2
   OR ( LEN (LI$) = 2 AND MID$
   (LI$,2,1) < > "(" )) THEN A$ =
   CHR$ ( ASC(A$) + 32 )
```

SAVE AUTHOR.HG.

You're done. My apologies.

INTERESTING OR UNUSUAL APPLICATION OF YOUR APPLE?

Call -A.P.P.L.E. wants to bring its readers the most interesting and unusual applications of Apple computers. If you think the way you use your Apple would be interesting to the rest of the club, please write and tell us about it. It could lead to a story published in Call -A.P.P.L.E. To let us know; A.P.P.L.E., Applications, 21246 68th Ave. S., Kent, WA. 98032.

UNCLASSIFIEDS

Members Non-commercial rate .05/word
Members commercial rate .25/word
All others: 1.00/word, $25 minimum
Send ad copy with payment to editorial department. Unclassified copy must be received by the fifteenth day of the month, two months preceding the date of issue. i.e., ads received by February 15 would appear in the April issue.

MICROMOVER quickly transfers files between Apple disks over a phone line, without conversion first. Data Compression algorithm. DOS 3.2 and 3.3. Hayes Micromodem II and acoustic modems accommodated. Only one copy required, as it transfers itself to distant end. Cost is $50.00 from DAB Systems, 9116 Rockefeller Lane, Springfield, VA 22153.

AIR NAV WORKSHOP (][+ ///e), fully-customizable, navigate any area, adjustable winds, automatic flight path replay, much more. $40. Check, MC/V, UPS-COD. SPACE-TIME ASSOCIATES, 2039-CA Country Club Drive, Manchester, NH 03102, (603) 625-1094, Free Info.

APPLE PUBLIC DOMAIN SOFTWARE FOR SALE. Seven Different Disk Volumes. Volume 1-7 for $12 each disk including shipping. All seven disks for $50. $1 for a catalog of all disk volumes. Send money order or check to: Gennaro Conte, 2522 Holland Ave., Bronx, NY 10467, Compuserve ID: 74005,1064 Delphi user name: CONTE or call me at (212) 798-2964.

DISCAT SAVES YOU TIME AND MONEY. It's an Apple-compatible menu-driven catalog on diskette for one-stop software shopping. The catalog contains a listing of over 750 items — at significant savings to you. Software is listed by name, publisher, type, or price range, or you can browse leisurely through the whole catalog at your own pace. Hurry! Gain access by booting up with DISCAT. Send $3 (Maryland residents add 5% state sales tax) to: Computer Ware Unlimited; Dept. CA; P.O. Box 1247; Columbia, MD 21044.

KEYNOTER. Specially designed to hold those valuable notes you need at the keyboard. We offer two types: flat overlay which fits around the keyboard or detachable clear plastic book which holds six 3 x 5 cards. Please specify which KEYNOTER + Apple //e or][+ when ordering. Send $9.95 + $1.50 shipping (CA res. add sales tax) to: Geisler Enterprises, 20333 Old Town Rd., Star Rte. #2, Tehachapi, CA 93561.

NOTCH-O PUNCH: Want to write on the back of your disk? Use our self-aligning Notch-o punch to add a write-protect notch. $8.95 + $1.05 shipping (CA res. add sales tax). Geisler Enterprises, 20333 Old Town Rd., Star Rte. #2, Tehachapi, CA 93561.

TIRED of arcade games? How about an adventure game that will be fun to play and make sense. We had trouble getting the game testers away from the computer! Hegira is a science fiction adventure game. It was written over three years by a professional programmer who has also published science fiction. Hegira — tale of a galactic exile requires a 48K Apple][,][+ or //e, with DOS 3.3. The mail price is just $15. We will pay US postage. Washington residents please add 7.6% tax. Send check or money order to Ted Butler 12545 SE 270th St. Kent, WA 98031. Warning — Hegira has proven to be habit forming.

COMPARE	THE FILER TO PFS	
Search 300 files	1 sec	3 min.
# of files on disk	4000	1000
PRICE	$50	$125

To order, mail $50.00 to DAN SCHMIDT 15841 Leadwell St, Van Nuys CA 91406.

3 UTILITIES FOR $50.00! LOCKER: Protect your program from copiers. FINDER: Find any commands, for fast editing. SHRINKER: Rewrites program; saves memory! To order, mail $50.00 to DAN SCHMIDT 15841 Leadwell St, Van Nuys CA 91406.

ALMOST WHOLESALE — Save BIG BUCKS on almost all Hardware & Software needs. Usually 20% to 45% below retail. Monthly specials. Send self-addresses stamped envelope for latest price list, or contact us to request quotes on specific items. COMPUTER PRODUCTS UNLIMITED, Apple Dept., 11116 East 46th Ave, CompuServe (73075,613) SOURCE (TCG249).

OBJ.APWRT][F Detailed, complete, and thorough disassembly script includes full details on customizing Applewriter //e and capturing your own source code. Two unlocked, crammed-full, double-sided diskettes plus a free bonus book for only $29.95. SOURCE-CODE, %Synergetics, Box 1300-AP, Thatcher, AZ 85552. (602) 428-4073. VISA accepted.

EDUCATIONAL SOFTWARE: Send for free price list of educational software for the Apple. The Country Store, P.O. Box 2131-Ced., Littleton, CO 80161 (303) 779-0883.

DISKINVOICE SYSTEM: Apple INV/AR software, perfect for YOUR business, only $55.00 COMPLETE! (Compare to $500 packages!). Sample printouts, details from: BROADWAY SOFTWARE, Suite 136, 642 Amsterdam, New York, NY 10025.

TIME ZONE AND MYSTERY HOUSE SOLUTIONS: includes maps and instructions. Time Zone $4. Mystery House $2. Both for $5D Contact Duane Allman 12805 W. Eden Trail New Berlin, WI 53151.

//e ABSOLUTE RESET to the "old" monitor. Get back in total control. User transparent till activated. No hole blasting. Passes diagnostics. Instructions & source-code on disk, plus "must have" bonus book $19.95. Old Reset, % SYNERGETICS, Box 1300-AP, Thatcher, AZ 85552 (602) 428-4073. VISA accepted.

MULTI-LINGUAL software for the Apple][+ or //e. FREE CATALOG. Le Professeur, 959 N.W. 53rd St., Ft. Lauderdale, FL 33309 (305) 771-6498.

NEW IDEAS and OLD FAVORITES: This month specials: WOW! SPEED-UP your VisiFile program — Finally! (I never thought it was possible) Famous LaFleur's FAST Report $35; FAST Sort $50; FAST Index $75. These programs make your VisiFile useable and perform its tasks much quicker. Special price for all three $149 or also sold individually.

APPLE /// — SOSTRAN is a general purpose utility to move files between][to /// or /// to][. Includes program, font, text, graphic files of all kinds. DOS, SOS or Pascal environments. $80.

DRIVER /// will allow printing thru any][parallel interface card. $30. PSCREEN allows printing of /// graphics or text screens to Epson, C.Itoh or Apple dot matrix. $50.

SUPER LOG QSL Ham Radio data base in ProDOS environment $29.95.

MAXELL quality disks: MD1-D,DD,SS was $46 now $29; MD2-DD,DS,DD 96tpi was $85 now $49. (please incl. phone; computer model)

For above products contact FORMULA 1 COMPUTERS, Suite 601 Madison Plaza 1115 Madison St. NE, Salem, OR 97303. Add $1 each item for shipping. VISA/MC/Amer.Express/Allow cks to clear. (503) 581-2687 msg.

APPLES'S APPRENTICE. . . The easy to understand Apple magazine for kids 8 to 80. Graphics, Comics, fun! 12 issue special introductory price $18. Write the secret word $4C on your check and get a 13th issue free! Emerald City Publishing PO Box 582-AA Santee, CA 92071

FOR SALE — MICRO-SPEED II + Forth Language System for Apple][. Superfast software/hardware (4 mhz C8231A arithmetic chip & board) implementation of '79 FIG Forth. Detailed reviews of the system in the Sept. '81 issue of *Call -A.P.P.L.E.* $350.—Bill White Box 276 Brewster, WA 98812 (509) 689-3367

RENT SOFTWARE $5 monthly. No membership fee. Send LARGE SASE to: Preowned Software 4229 Baker Ave NW Seattle, WA 98107 for list.

APPLE /// COMPUTER, like new! Barely used, with AP3 monitor, 256K, all system software. Only $1899. Call Howard Rothman after 5pm EST: (203) 333-6436.

APPLE IEEE-488 INTERFACE CARD. Apple product No. A2B0015. manual and original box. — $225 — Jim White 17511 26th Dr SE Bothell WA (206) 481-7507

700 PROGRAMS & MORE: Over 700 programs available for business, education, and games for the][and //e $8. Discounted peripheral cards. Details: Reliant PO Box 33610 Sheungwan Hong Kong.

FOR SALE, WORDSTAR 3.30, $178, also Applewriter II, $48, both never used or even opened, Ralph Smith, day: (714) 632-3024; night: 544-4586, or write: 13572 Flint Drive, Santa Ana, CA 92705.

Apple Announces Lisa, Mac Seminars

APPLE Computer, Inc. announces a one-day and week-long seminars for hardware and software developers to introduce the variety of development environments for Macintosh and Lisa. Subjects to be covered include product features and positioning, marketing strategy, development and operating environments, and development tools.

The one-day seminar will be open to principals and engineers of companies actively involved in software and hardware development for Macintosh and Lisa. It will be held on Monday, April 23 at the Grand Hyatt, New York, NY.

Fee $75 (includes seminar and noon meal); Limited to two persons per company; To make reservations and for more information, contact: Dennis Gallinat, Developer Relations, Apple Computer 20525 Mariani Ave., Mail Stop 23-AF Cupertino, CA 95014

Intensive week-long technical seminars for software developers will provide in-depth instruction on how to use the Macintosh and Lisa development tools. They will be open to experienced Pascal programmers actively involved in software development on Macintosh *and* Lisa.

May 14-18 Palo Alto, CA
(exact location TBA)
June 18-22 Boston, MA
(exact location TBA)

Development tools that will be covered include Lisa Workshop, QuickPort, and the Macintosh ToolBox. Instructors will be members of the Apple 32 Systems Technical staff. For more information, contact Dennis Gallinat, as above.

—A.P.P.L.E. BITS & PIECES

APRIL 1984

MEMBER BILLBOARD

Feedback

FEEDBACK, sometimes known as "groans and gripes," is important to us. With the advent of the Member Billboard column, this becomes the ideal arena in which to air problems, provide solutions and/or explanations, and to pass on miscellaneous bits & pieces of club information. We will do our best each time to answer member's questions that may be of interest to others, and to offer our members a look at our operations.

We received a letter dated February 1st, which read:

This is an irate member letter.

My Call -A.P.P.L.E.s arrive at my home a full two months after *they appear on my computer dealer's shelf. This is outrageous.*

The issue which arrived at my home today *is the January 1984 issue!! My dealer had both the January and February 1984 issues on his racks last Tuesday (January 24, 1984). Why?*

We agree it's outrageous, and we can't think of a better word to describe the situation. Dealer copies are shipped via UPS, and typically reach their destination within three or four days. Unfortunately, members are at the mercies of the postal service, as are the readers of every other magazine, and it is not at all uncommon for third class mail to take three to five weeks to be delivered. Your first step would be to contact your local post office and ask them why you are not getting better service. Another factor that may be involved in your case is the size of the town in which you live. The postal regulations for bulk mailing provide that any zipcode bundle that has less than ten pieces be placed in a general state or area bundle.

We have applied to the post office for a second class mailing permit, which usually takes a few months to process. We have done so on the assumption that, as the post office has indicated, second class mail receives more expeditious handling.

Lastly, for the very reasons of member frustration, as you have so clearly written, we have, almost from the beginning, offered first class mail to our members for an additional $7.50 per year. While more expensive, it is at least one way to circumvent slow delivery. First class usually reaches members three to four days from the date of mailing.

We often receive inquiries from overseas as to why we cannot ship hardware to our foreign members. This is a problem that has disturbed us for quite some time, and we are still looking for solutions. The red tape in shipping for export would make an interesting article itself, but in brief, the requirements, as we understand them, are that we must apply for an individual export license for each product we plan to ship, *for each destination country,* thus to license the 35 or so hardware products we carry for export to the over 70 countries we serve, would require over 2400 individual applications and fees. It is likely the charges we would have to assess would offset the original discounts we offer.

We will continue to explore whatever possibilities are open to make our hardware available overseas. For those of you with friends or relatives residing in the United States, it is possible for us to ship to them in your behalf. Export requirements for individuals, as opposed to businesses, are far less stringent, and they may take a package direct to a customs broker or air freight organization for shipping. If this solution applies to you, all you need to do is indicate a U.S. shipping address on your order.

Time Zones are Tricky

Sharp-eyed reader Ken Crowe (and others) point out we goofed up the timezones in the March *Billboard,* When it's noon, eastern, our offices (at 9 am pacific) are just opening, and at 4 pm pacific time, east coast folks are probably eating a 7 o'clock dinner. Apologies!

System 38: a Shot in the Arm

Right now, we might well be dead in the water without our IBM (who?) System 38 mini-computer. Indeed, the 38 was a real "shot in the arm." Early last year, with our membership around 14000, printing labels to mail each issue of *Call -A.P.P.L.E.* often took two to three working days. The Apple][is a fine computer, but it — and any other micro or personal computer — has its limitations. Using the same example, we now run magazine labels for some 20000 members in about five hours.

Prior to the implementation of the 38, we had no way to check on member orders other than manually pouring through reams of order forms, while now an inquiry, and virtually instantaneous response, can be made from one of the eight on-line terminals, or through an Apple ///, which is set up to emulate a DEC VT-100 terminal.

The design goal for the System 38 is a complete, integrated data base system, with three major modules:

member maintenance
order entry
inventory control

The first two have been accomplished for some time now and when inventory control is brought on-line shortly, each item of an order that is entered will deduct the product from inventory and, when appropriate, purchasing will be advised on low inventory conditions.

29 year old **Mike Christensen** is a systems analyst and project leader for the System 38, but his responsibilities are shared equally with 23 year old **Maureen Walsh,** a Gonzaga graduate who functions as programmer/analyst and system operator, and systems analyst **Garry Reinhardt,** who brings us an 18 year background of IBM systems.

Working closely with the System 38 group are two other individuals, **Pat Lindsay,** who in her function as a member service representative, handles a great deal of member correspondence and order inquiry calls, and **Gunter Hirt,** one of our earliest employees, who is responsible for assigning all new product codes and prices, and verification of all order "batches" before they are handed on to shipping.

We hope this has given you somewhat of an insight into the vital functions played by the System 38 in our day-to-day operations. The 38 has a current but expandable capacity of about 50000 members, and a number of different ideas are under consideration for additional uses that would provide you with improved and new services.

A.P.P.L.E. Aids Blind

patrick schwab

SINCE 1979 the annual bowling tournament for the Northwest International Blind Bowling Association (NWIBBA) has been run with the help of an Apple][+ computer. The September tournament brings over 75 blind individuals together from the Pacific Northwest, British Columbia and California to participate in this annual event. The Apple is used primarily to compute the winners for the event. The NWIBBA awards over fifty trophies for three events in three separate divisions. The statistics have to be completed just three hours after the last ball is thrown for the awards banquet. Hence we have a real need to get the winner's list out quickly and accurately.

The computer has become a crucial component to the successful operation of the NWIBBA Tournament. A.P.P.L.E. played a vital role in the 1983 Tournament held last September. As the executive secretary of the NWIBBA I have been bringing my Apple to the Tournament since 1979. When I arrived in Seattle Friday night from Salem, Oregon, I realized I had the computer, the drives, and the printer but where was the monitor? Yes, I had left the monitor 230 miles away in Salem. My only resource at 8:30pm was a copy of *Call–A.P.P.L.E.* (Everyone carries a copy of the best Apple magazine for such emergencies, don't they?)

I called Val Golding on the hotline and explained my plight. He saw no dilemma. Call Dick Hubert, he said, a monitor could be lent out of the Club's facility in Kent. Dick was as cooperative as could be. I met Dick early on Saturday at the A.P.P.L.E. offices. He produced a NEC monitor and on my word alone that I was a club member I carried it out the door and headed for the bowling alley. Because of Val and Dick's effort, the Tournament went off flawlessly. This was an advantage of membership I never would have anticipated.

The software to run the Tournament had to be written from the ground up. The specifications seemed too unique in 1979 for any commercial software. We have three divisions, an A division, B division and Auxiliary. The Auxiliary division is the sighted bowlers who belong to the individual blind bowling leagues. (Someone has to tell the bowler they left a 7–10 split and keep score.) The A and B divisions are determined by average. The high average bowlers are in A and the lower bowlers are in B. This gives the lower bowlers a chance at scratch (scores without a handicap added in) trophies.

The software had to calculate the handicap and the division. It had to identify doubles partners as well as team members. After the scores were typed in, it had to calculate all the different events, for the three divisions, sort for the winners and print more than fifteen reports. Needless to say the task by hand is quite laborious. With the computer we are finished 45 minutes after the last ball is thrown.

The programs (there are over 10 of them) are written in Applesoft with a small assembly language shell–metzger sort to bring the high scores to the top of the list. The edit routines are screen oriented to allow for on screen changes of individual records. The entire system can run on one disk drive for up to 300 bowlers. A printer is optional.

This program was begun back in ancient history when I had one drive, a TV/RF modulator, 32K and no printer. The program has grown to require 48K, but it still uses a 40 column display. I would have included a program listing in this article, but it is much too long.

The program has not been without its problems. The biggest obstacle has been with the calculation of the handicap. The handicap the NWIBBA uses is 90% of the difference between 180 and the bowlers' average. The results must be converted to an integer and not rounded. Easy you say! Consider the following formula:

```
100  HD = INT(( 180 - AV ) * .9 )
```

AV is the variable for the average and the results (HD) should be the handicap. I didn't think I would suffer the indignities of Apple's rounding error, but alas the above formula gives me proper results when there are numbers to the right of the decimal. Such as:

```
94    = INT(( 180 - 75 ) * .9 )
```

But if the results are already a whole number it magically rounds up, as in the following:

```
73    = INT(( 180 - 100 ) * .9 )
```

Work that one out with your RPN calculator — the results should be 72. I am not here to explain the phenomenon. I solved the problem by using a look–up table.

Other than a few frustrating moments during the first couple of tournaments, the program was in service and our Apple has served our needs quite well.

How Do Blind People Bowl

For the readers that are wondering just how a blind individual would bowl, allow me this quick explanation. First, over half of the bowlers have usable visual acuity. That is, legal blindness is defined as 20/200 visual acuity. (They can read the big E at 20 feet but not the rest of the chart.) These folks usually have no problem bowling with no physical adaptations. The bowlers

with little or no usable vision use a bowling rail. The rail is constructed of light aluminum tubing. The rail is three feet off the ground and about twelve feet long. To weight the rail down four bowling balls are placed in the base to keep it in place.

A bowler using a rail delivers the ball in the normal three, four or five step approach except the free hand slides along the rail. This keeps the bowler straight and clues as to when to release the ball.

The scores in the NWIBBA Tournament rivals any tournament. At the 1983 Tournament held in Seattle some scratch scores were over 200. That is respectable for anyone, sighted or blind.

Chairman of the Board

david l pontius

AFTER reading many reviews of a popular home accounting program, I thought it would be an excellent addition to our household data processing department. The next step would be to influence the Chairman of the Board, of our household (my wife), that this program was a necessary purchase. With that thought in mind I fired off an inner-departmental memo to the chairperson. She replied through the usual organizational channels, her official letterhead (a note taped on the refrigerator). She reminded me of our last board meeting concerning REAGANOMICS, and how the Board-of-Directors had limited the data processing department's budget to ten ($10) discretionary spending (i.e. magazines, posters, and or stickers). The memo also brought to my attention the organizational policy of the management, which was to request a requisition of all purchases of $10 or more from my department, before the Chairperson would consider authorizing a purchase order. In corporate circles I had been stone walled by the Board of Directors. Being a minor functionary in the bureaucratic organization of running a household, I did not get discouraged. I love the opportunity to "play hard ball" with the top management.

The only corporate executive strategy left to me was to influence the Board of Directors. This was the point in the corporate game where I had to rely on my many years of corporate finesse. With my executive career on the line, I had to gamble with the entire data processing budget. I boldly issued a cash voucher to myself for the entire ten dollars (all or nothing). My next move was critical, find a babysitter. This move cost 30% of the entire budget, but there was still enough funding left to pull off this last effort. I invited the Chairperson out for lunch, hoping that plying her with a hamburger and diet soda would weaken her stand on the current household procurement policy. Casually during lunch, I happened to mention that there was a computer store just around the corner and that they would be glad to demonstrate a popular home accounting program, at no charge or obligation (the last part got her attention). She reflected on the possibility for an eternity (perhaps two or three minutes). The whole time she was reflecting, I was struggling to maintain my executive composure (i.e. cool, calm, disinterested). When this eternity finally collapsed, the Chairperson consented to a brief demonstration.

Upon arriving at the computer store, I was again briefed on household organizational procurement policy, and the pitfalls of REAGANOMICS. As I meekly followed in the wake of the Chairperson, the salesrep greeted us. At this point in time, I had everything on the line. The entire data processing budget was committed, with a possible over-run because of the babysitter time element being miscalculated. The entire corporate strategy now rested in the hands of the salesrep. Does he have the ability to face the Chairperson? Does he understand the power-struggle involved? Did he go to business school? The pressure was mounting as these questions raced through my mind. Being numbed from the pressure, thinking of the budget over-run, I was not following the demonstration.

Evidently the demo went quite well, the Chairperson was impressed with the control that this program could give management over budgeting, cash control, and trend analysis, I was purged from my comatose state, by the mention of an emergency boardmeeting. This meeting actually only took a few minutes, the contingencies of the purchase were explained, in detail to me, as options:

1) If I were not capable of operating the program as demonstrated:
   a. She would personally freeze my assets.
   b. I would be grounded for life.
   c. I could be transferred to another household organization, and my data processing department spun-off to a hostile merger,

Or

2) If the home accounting program worked as explained,
   a. I could get a promotion.
   b. The data processing budget would be increased.
   c. The Chairperson would be in charge of using the program exclusively.

Our household did purchase the program, I since have been promoted to Vice President in charge of Maintenance, Operations, and Service. The Chairperson (my wife) now is in charge of data processing, with a larger budget.

Success is all in how you play the corporate game.

AN INTRODUCTION TO STRUCTURED PROGRAMMING:

Pascal Low-Res in the High School

ian jackson

THERE must be many factors that lead to the success of the Apple][as a personal computer, but almost certainly a major one was its graphics capabilities.

The use of graphics as a teaching and learning tool is extensive. (Throughout the rest of this article I will use the term teaching to include the idea of self teaching or learning) Most books that attempt to teach about the Apple, very quickly get into the use of graphics. The immediate feedback of the "you see what you've done" approach is a powerful reinforcer in the learning process. Much of the success of Apple Logo must be attributed to this.

Well, where does this leave us when we want to teach structured programming in Pascal on the Apple? The Turtle Graphics Unit provided with Apple Pascal is a sophisticated graphics package, and this is its downfall as far as the beginner is concerned. It is really a case of just too much at once.

First, the student has to come to grips with a powerful operating system. Experience with BASIC or Apple DOS is not a great deal of help. The editor is a good example. It has few peers even on large systems, but is much more complex than the Apple's normal simple editing facilities. For example, a common problem is not realizing that the editor is a tool, and that a basic understanding of it does not bring a student any nearer to the concepts of structured Pascal programming, even though it must be mastered to use the Apple Pascal language system.

Second, some knowledge of program structure and Pascal syntax is required. Syntax errors, with their, at times, less than illuminating messages, can be enough to frustrate even an experienced programmer. For the beginner, it can give a yoyo feeling with a never ending cycle of compiler error, editor, compiler error, editor and so on. For most, Pascal is best attempted with an understanding and experienced Pascal programmer no further away than the other end of the phone.

If Turtle Graphics is added to this so that the "you see what you've done" approach can be used, it becomes a final nail in Pascal's coffin as the student is driven or seduced back to the friendly, interactive, though messy (in terms of structured programming) BASIC interpreter.

Bearing in mind the above, I turned to the Apple's much under-rated low resolution graphics capabilities and produced a library unit called LORES to implement an Apple Pascal version. The idea was to produce a simple graphics unit for beginners to use as a learning tool. The procedures are logical and lend themselves to structured programming techniques. For example, true cartesian coordinates (with the origin always in the bottom left corner of the graphics screen) are supported in both full screen graphics and mixed graphics/text mode.

The unit as a whole is user-friendly and very forgiving. It is unlikely that a runtime error will occur due to a mistake in the program. This is important, since there are none of the Applesoft "'beep' ILLEGAL QUANTITY ERROR IN LINE XXX" type messages, if a runtime error occurs in Pascal. Such errors cause the whole system to "fall over" with a cryptic error message (such as "value range error") and a mixture of letters, symbols, and numbers. (To the beginner, such a message along with "S# 1, P# 2, I# 56," can be less than enlightening.) The system must then be reinitialized or rebooted. This sort of problem can be very disheartening for the beginner, who often has little idea of its cause.

Finally, the unit is useful for introducing the concept of procedures and their parameters and later, the idea of data types.

LORES consists mainly of four machine code procedures. These are linked to a Pascal skeleton to form the intrinsic unit, since machine code procedures and functions can not form this type of unit. For those unfamiliar with Pascal, a unit is a subprogram that can be used as if it was part of the host program. The code of a regular unit is linked to the host program and its code copied into the codefile of each host. The advantage of an intrinsic unit is that it resides in the SYSTEM. LIBRARY and is loaded into memory as required by a host program. In this way, several programs can use the same copy of the unit.

To improve the logic and the user friendliness, LORES defines two new data types. The first "scrntype" has only two values, mixed or fullscrn, and is used to control the type of graphics display. The second "lrcolor" has for its values the standard Apple low resolution graphics colors.

LORES's four procedures, as can be seen in the interface part of the unit, are:

```
graphics ( display:scrntype );
setcolor ( color:lrcolor );
plot (x,y : integer );
normal;
```

The first procedure (graphics(display)) sets the low resolution graphics mode to mixed or full screen graphics, depending on the value of the variable "display" and clears the graphics screen to black. The plotting color is set to black. A beginner might think white a better choice so

that plotting can begin immediately, but an important principle is that variables (including color plotting ones) should be initialized to the desired value by the user. The procedure employs a much faster graphics screen clearing routine than the routine in the F8 ROM, which, with its forty calls to the subroutine VLINE, seems to have sacrificed speed for a saving in code.

Setcolor(color) does what one would expect. It sets the color for subsequent plotting according to the value of the color parameter. It is simpler to use than the BASIC version. For example, "setcolor-(orange)" should be easier for the beginner than "COLOR = 9".

Plot(x,y) plots a point at the true x,y location on the screen. The two parameters "x" and "y" can take any integer value as "x mod 40" and "y mod 40 or 48" are used. As can be seen in the "Fly Ship" sample program, wrap around occurs. Using negative integers is also allowed, since the absolute values of the parameters are taken by the machine code routine. This can be used by more advanced programmers to give effects such as in the "Fly Ship" sample program, where, if the ship is given a negative starting X coordinate, it flies right to left instead of left to right.

The final procedure, normal, resets the display for text, then clears the screen and homes the cursor.

To use LORES, it is best to assemble, compile, and link and then install the resulting codefile in the SYSTEM.LIBRARY. Although this all sounds a little daunting for the novice, the Pascal Operating System Reference Manual (p.ref) is fairly easy to read and is particularly good at pointing out the method to follow with a one drive system. The following instructions assume that you know how to write, type in, compile, and execute a short Pascal program.

1) Type the assembly language routines into the Pascal editor and save it with the file name GR
2) Assemble the file. (instructions and files needed in p.ref page 136) When asked the question "TO WHAT CODEFILE?" answer GR. CODE[6]. This will avoid a possible "no room on volume error" occurring.
3) If you are not using the original SYSTEM.LIBRARY provided

10382 STANFORD AVE.
SUITE E-203
GARDEN GROVE, CA 92640
PHONE: (714) 534-4216

Laing Electronics

MICROSOFT 16K CARD
$59.00

★ HARDWARE FOR THE APPLE II ★

	LIST	YOUR COST
HAYES MICROMODEM II	379.00	258.00
HAYES SMARTMODEM 300 BAUD	289.00	199.00
HAYES SMARTMODEM 1200 BAUD	699.00	498.00
NOVATION APPLE CAT II	389.00	268.00
MICROSOFT Z-80 SOFTCARD CP/M	345.00	219.00
MICROSOFT PREMIUM PACKAGE (Z-80, VIDEX 80 COL., 16K CARD)	695.00	490.00
PCPI 6 Mhz APPLI-CARD (Z-80 CP/M)	375.00	279.00
VIDEX VIDEOTERM 80 COLUMN CARD (SOFTSWITCH, AND INV. CHAR. SET)	319.00	225.00
WIZARD 80 COLUMN CARD	249.00	149.00
WIZARD BUFFERED PINTER INTERFACE (16K BUFF + GRAPHICS)	179.00	129.00
(32K BUFF + GRAPHICS)	219.00	153.00
NEC HI-RES GREEN MONITOR (JB1201)	285.00	153.00
MICRO-SCI A2 APPLE COMPATIABLE DISK DRIVE W/O CONTROLLER	345.00	229.00
MICRO-SCI A2 DRIVE WITH CONTROLLER	445.00	299.00
OKIDATA MICROLINE 92/P PRINTER	699.00	498.00
PROWRITER 8510AP PRINTER	495.00	349.00
KENSINGTON SYSTEM SAVER FAN	89.95	65.00
GIBSON LIGHT PEN/LPS II (PENTRACK)	349.00	239.00
VISTA V1200 CART. DISK SYSTEM 6.2MB	1840.00	999.00
VISTA A-800 8" DISK CONTROLLER	379.00	249.00
VISTA SOLO 5¼" DISK DRIVE	299.00	195.00
8" DRIVE SUBSYSTEM WITH CONTROLLER 2.4MB	1900.00	995.00

★ VERBATIM DISKS ★

100 DATALIFE 5¼" (MD525-01-18158)	490.00	239.00
50 DATALIFE 5¼" (MD525-01-18158)	245.00	124.00

★ SOFTWARE FOR THE APPLE II ★

PFS: FILE (SOFTWARE PUB.)	125.00	76.00
PFS: REPORT (SOFTWARE PUB.)	125.00	76.00
PFS: GRAPH (SOFTWARE PUB.)	125.00	76.00
DB MASTER REG. VERSION (STONEWARE)	229.00	152.00
DB MASTER VERSION #4 (STONEWARE)	350.00	231.00
MULTIPLAN (MICROSOFT) CP/M OR DOS3.3	275.00	167.00
WORDSTAR (MICRO-PRO CP/M)	395.00	327.00
WORDSTAR WITH Z-CARD (PCPI APPLI-CARD)	495.00	425.00
SCREEN WRITER II (ON-LINE)	129.95	86.00
FORTRAN-80 (MICROSOFT CP/M)	195.00	129.00
ASCII EXPRESS PROFESSIONAL	129.95	85.00
Z-TERM PRO. (SOUTHWESTERN DATA)	149.95	98.00

If Personal Check, Allow
10 BANKING DAYS for CLEARANCE.
Prices and specifications subject to change without notice. Unless otherwise noted,
ALL PRICES are PLUS SHIPPING.
SHIPPING & HANDLING—ADD $5.00 UPS SURFACE
(DISK) DRIVES $7.50—MONITORS $10.00—
PRINTERS $15.00.
California residents add 6% Sales Tax.

Circle 22 on Reader Service Card

with the Pascal System, use the utility LIBMAP (p.ref page 194) to find which segments in your system library are already in use.

4) Type in the unit LORES as you would a normal Pascal program, leaving out the comments ("{..to .. }"). Use a segment number of "24" with the original SYSTEM. LIBRARY or an unused segment# if this one has been used. Segment number "24" has been used in the listing (hence the "intrinsic code 24").
5) Compile LORES to the codefile LORES.CODE.
6) Link GR to LORES using the Linker (p.ref page 176). In addition to SYSTEM.PASCAL and SYSTEM.LIBRARY (which should already be on your boot disk) you will require SYSTEM. LINKER, LORES.CODE, and GR.CODE, on your boot disk if possible. (If LORES.CODE or GR.CODE are not on this disk, you will have to prefix their names with #5: or whatever.)

Proceed as follows:

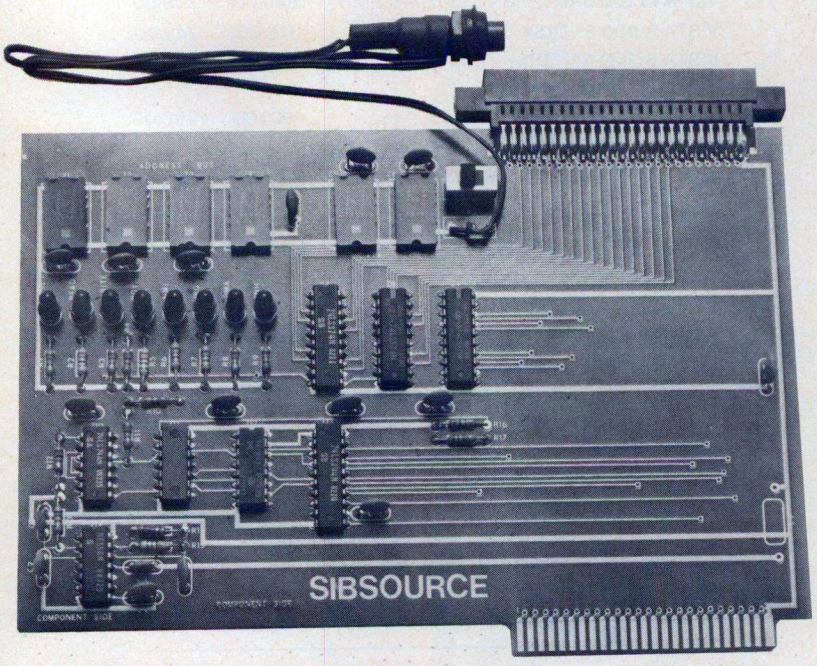
a) from command level press "L" for the Linker
b) for the "HOST FILE?" prompt type in LORES
c) for the "LIB FILE?" prompt type in GR
d) for the next "LIB FILE?" prompt press return.
e) for the "MAP FILE?" prompt press return.
f) for the "OUTPUT FILE?" prompt type in LORESUNIT

7) Install LORES in the SYSTEM. LIBRARY. To do this, use the system librarian utility (LIBRARY. CODE p.ref page 186). It is probably easier to have the needed files on you boot disk. These are: SYSTEM.LIBRARY and LORESUNIT.CODE. Now eXecute the file APPLE3: LIBRARY. CODE and proceed as follows. (after restoring boot disk to drive 1 with a one drive system)

a) for the prompt "OUTPUT CODE FILE – >" type in NEWLIB
b) for the prompt "LINK CODE FILE – >" type in SYSTEM. LIBRARY
c) press the "=" to copy all the existing units into the new library.
d) press "N" to select a new file then to the "LINK CODE FILE – >" prompt, type in LORESUNIT
e) LORES will be in slot 1 of the link code file so press "1" then the space bar followed by a vacant slot# in NEWLIB and then the space bar again, to copy LORES into NEWLIB. (slot# 6 onwards is vacant in NEWLIB if it was copied from the original SYSTEM.LIBRARY)
f) Type "Q" to quit and then press return after the prompt "NOTICE?".

8) Finally use the FILER to change the name of the existing SYSTEM. LIBRARY to OLDLIB, and NEW-

APPLE BUS MONITOR & EXTENDER



BUS MONITOR – ADDRESS and DATA BUS are displayed simultaneously in Hex. LEDs show READ, WRITE, RESET, IRQ, NMI, RDY, INH and DMA control lines.

CONTROLS COMPUTER SPEED – Allows normal, slow, single step and halt of any programs on any operating system without software changes. Undetectable by software.

A SLOT-INDEPENDENT EXTENDER CARD to help in developing or troubleshooting any Apple compatible cards.

SEE INDIVIDUAL INSTRUCTIONS as any program executes. Valuable as a learning tool or to find that elusive software problem.

ALLOWS YOU TO DO THINGS that you would not be able to do without the use of expensive logic analyzers.

DEALER INQUIRIES INVITED.

SIBSOURCE INC.

15735 Nordhoff St. #52 • Sepulveda, CA 91343
(818) 904-1219

Order by phone or mail.
By mail send check or money order for $225.00.
Calif. residents add 6.5% sales tax.
Add $3.00 shipping.

Circle 23 on Reader Service Card

LIB to SYSTEM.LIBRARY. LORES is now installed and ready to use in programs.

Structured programming does not require the use of structure diagrams, but when I look at some of my earlier large Pascal programs, or more importantly, try to alter them, I wish I had learned to use structure diagrams first. The two most important features of these diagrams are that they encourage a top down design (or step-wise refinement) approach to programming, and force the programmer to relate the various parts of the program to each other before any code is written. Well structured programs are easier to maintain and debug, and are also more easily understood by other programmers (and the author six months hence!).

The program examples are relatively trivial, but they will serve to demonstrate the use of LORES and structure diagrams.

1) **Program Randomwalk.** Plot a point in the center of the screen and then move randomly one square, up, down, left, or right until the next plotted point would be off the screen.

The diagram is read from top to bottom and left to right. In this example you read from the main program heading to INITIALIZE and then read from left to right the statements comprising this procedure. In other programs, some or all of these statements may themselves be procedures or functions. Each of these, if detailed, would be read in the same fashion before proceeding to the next statement. After INITIALIZE, the next part of the program is the repetition statement. This particular one is "Repeat . . . Until a condition is met," but Pascal also has "While condition is met do. . . ." and a "For" statement (similar to BASICs FOR.. NEXT). Each of the statements connected to the repetition statement, are executed in the normal left to right order, repeatedly until (in this case) it's not ok to plot the next square.

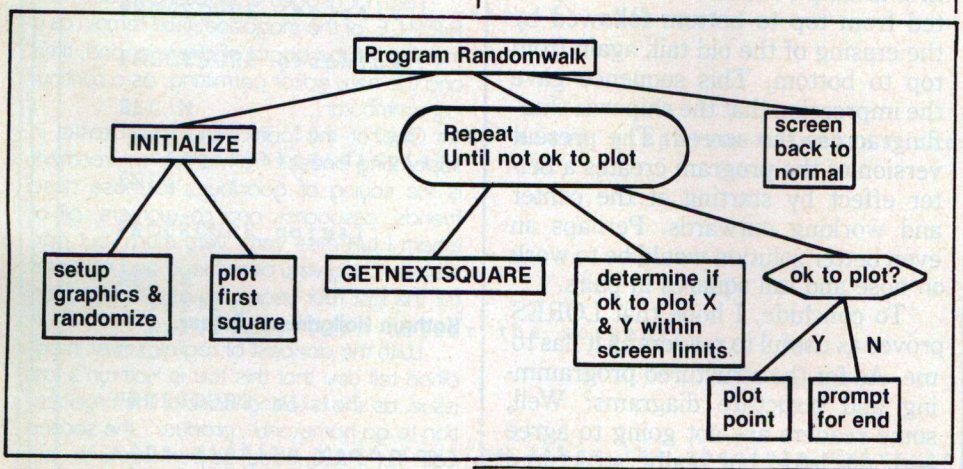
Usually a structure diagram stops short of detailing code, but in some cases such detail may be desirable. eg.(a) In the procedure INITIALIZE the statement "plot 1st square" could be detailed as:

The amount of detail in this diagram is restricted by the size of the page. A much better technique is to use the back of a long piece of printer paper, preferably 132 column (used 132 column printer paper is often available free from businesses that use computer printouts extensively). More than enough room is available for most programs across a five foot piece of this!

When the program is written, indentation is used to reflect the vertical level of the statements on the structure diagram. As a general rule, I use a two space indentation per level within a procedure. (I also use lower case for all program text, but in the listing I have changed some key words to upper case to make them easier to follow.)

2) **Program Flyship.** Move a simple space ship shaped object across the screen by plotting in the new squares and plotting out the old squares.

This is a common technique for moving a shape as it gives much smoother movement than completely erasing the shape and redrawing it.



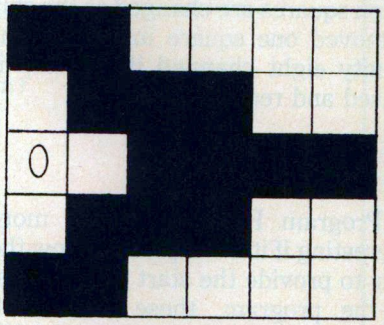


Diagram (A) The ship.

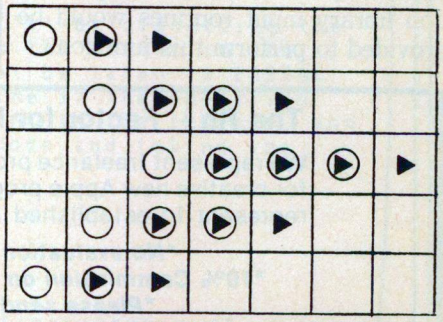
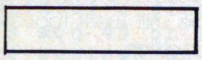
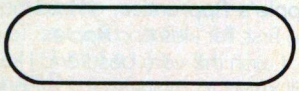


Diagram (B) The ship replotted.

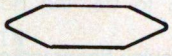
In my structure diagrams:



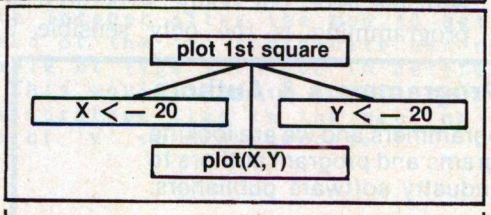
shapes contain some kind of action to be taken



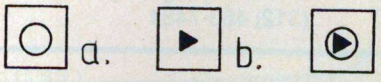
shapes contain repetition statements



shapes contain decision statements



The " < — " symbol stands for "becomes." eg. "X becomes 20." In Pascal " := " is used as the least confusing approximation to " < — ". eg.(b) procedure GETNEXTSQUARE could also be detailed.



Many fail-safe features are included. The data are safely stored on disk at all times, with the exception of the one new set just being entered during data entry sessions and during conversion/manipulation of data. If the prudent data disk back-up recommendations are followed, the user should not stand any risk of losing any data, which would need to be manually re-entered. The user can backstep in the program or opt out by "Pressing Q for Quit" in numerous places. In addition, pressing the Reset key returns the user to the beginning of each particular section.

NUMBER COMMANDER™ is written and operates almost entirely in Applesoft Basic. All rights to the code, including the copyright, belong to ComputerEase, Inc. Appleton, Wisconsin, who licenses its use to the customer. The code is open so that the experienced programmer can inspect, interpret and amplify it for his/her own personal use. Both program disks are perfectly copyable by standard copy programs.

NUMBER COMMANDER™ was conceived as a user friendly, general purpose program for engineers and scientists, not as a comprehensive statistical or other special purpose program, of which there are quite a few on the market. Hence, NUMBER COMMANDER™ doesn't do everything.

Hardware

Several popular printers are supported at the press of a key. A special section of NUMBER COMMANDER™ takes the inexperienced user through all the necessary steps of defining control characters for operation of other printer/interface card combinations.

Two disk drives are required as is an 80-column board. Apple Computer, Inc's 80-column board for the Apple *II*e works fine as does the Smarterm for Apple models][Plus and *II*e. Other compatible 80-column boards should work just as well.

This is a truly professional and very user friendly package aimed at professionals in engineering and science — not at the computer specialist.

NEW EDUCATIONAL SOFTWARE

The Learning Company, producer of award-winning educational software packages, has introduced five new discovery learning programs. . . Reader Rabbit and The Fabulous Word Factory, Word Spinner, Number Stumper, Addition Magician and Colorasaurus.

With these new offerings, The Learning Company product line has two new reading programs, two new number programs and its first art program.

Reading Programs

Reader Rabbit and The Fabulous Word Factory is a completely new approach to teaching essential pre-reading and early reading skills to children aged 5 to 7. By helping Reader Rabbit match pictures and words, label boxes and load words into a word train, children learn to recognize and spell more than 200 three-letter words. The program sells for $39.95.

Word Spinner takes young readers, aged 6 to 10, on a whirl through the alphabet to learn the building blocks of reading. In building more than 500 three-letter words and 1,000 four-letter words, children learn to recognize word patterns and develop critical vocabulary and spelling skills. Emphasis is placed on proper positioning of vowels and consonants to enhance word pattern recognition. Priced at $34.95, the program runs on Apple][computers and is available now.

Number Programs

Number Stumper, the computer-age version of a classic 14th century game of chance, teaches basic addition and subtraction. Using animated dice, children aged 6 to 10 create and solve mathematical equations and develop abstract reasoning and strategic thinking skills. The program sells for $39.95.

Addition Magician is a fast-paced, race-against-the-clock number strategy game that teaches children the basic concepts of addition and flexible thinking about numbers. Children build magic walls around groups of numbers that add up to a target sum. The program, for children aged 6 to 10, is priced at $34.95.

With the five additions, The Learning Company now offers 14 educational software packages. Its other programs are Juggles' Rainbow, Bumble Games, Bumble Plot, Gertrude's Puzzles, Gertrude's Secrets, Moptown Parade, Moptown Hotel, Magic Spells and the popular Rocky's Boots, called a "software classic" by The New York Times.

The Learning Company is headquartered at 545 Middlefield Road, Suite 170, Menlo Park, CA 94025. Telephone 415/328-5410.

CLASSIC COURTROOM SIMULATION

Navic introduced JURY TRIAL II — the courtroom strategy game — at the Softcon show in New Orleans. JURY TRIAL II features an enriched cast of characters supported by individual dossiers in the accompanying documentation. Among the colorful new personalities introduced in the advanced version are: Mr. Wallbanger, the town drunk, and the voluptuous Linda LaVerne — a flirtatious night club dancer.

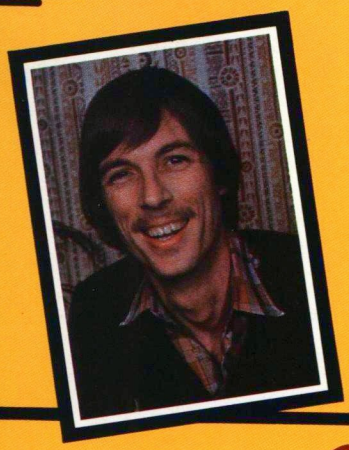
Players in their roles of prosecutor and defense attorney, attempt to "stack the jury" by choosing candidates biased in favor of their own respective sides, but even though the jurors exhibit favoritism, the trial is won or lost by the skillful tactics of the competing attorneys.

Case evidence is developed by interviewing the venerable County Sheriff, who is rather uncooperative, impatient and generally onerous to deal with.

During the trial, the computer acts as Judge — controlling the proceedings, ruling on objections, admitting evidence, and occasionally citing a witness for contempt of court. If, after the case is tried, the jury can agree on a verdict, the "Judge" will pass the appropriate sentence, or set the defendant free. Appeals to a higher court are possible, but bail must be posted.

Packaged in a handsome gold-inlaid textured black vinyl case, JURY TRIAL II is priced at $49.

For additional information contact Vic Ratner (305) 627-4132. NAVIC SOFTWARE — North Palm Beach, FL 33408



southwestern data systems™

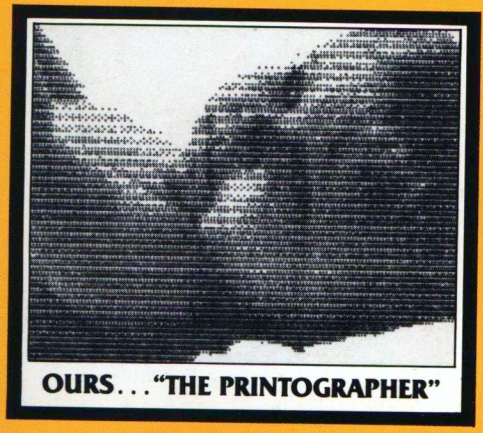
CHANGES ITS NAME TO

Roger Wagner Publishing, Inc.

With my name on the software, you have my personal guarantee of uncompromising quality, ease of use, and reliability in both product and support.

—Roger Wagner, President

First Compare, Then Demand It By Name!



OURS . . . "THE PRINTOGRAPHER"

THEIRS . . . "ZOOM GRAFIX"

Zoom Grafix is a trademark of Phoenix Software, Inc.
Apple is a trademark of Apple Computer, Inc.

"THE PRINTOGRAPHER"

If you have a printer and a printer card, Printographer is all you need to print any picture from your Apple's screen. No need for expensive graphics printer cards or other devices, Printographer simply offers the most features, and the best value. Compare for yourself. With Printographer, it's easy to see the difference.

FEATURES	THEIRS	OURS
Hi-Res and Lo-Res Printing	No	Yes
Reg or Inverse Inking	Yes	Yes
Picture Positioning	Yes	Yes
Magnification	Yes	Yes
Pause Printer	Yes	Yes
Color Printing	No	Yes
Add Text to Picture	No	Yes
Install in your Program	No	Yes
Completely Visual Editing	No	Yes
Oval Editing	No	Yes
Diamond Editing	No	Yes
Auto Picture Save	No	Yes
Compressed Pic Load/Save	No	Yes
Printers Supported	33	42
Interface Cards	34	34
Copyable	No	Yes
Backups	$10.00	Yes
PRICE:	**$49.95**	**$39.95**

Printographer supports virtually ALL printers including Apple Image Writer and Dot Matrix, C Itoh, Epson, Gemini, IDS, NEC, Okidata, Diablo, Qume, and Color printers such as Prism, Transtar, and C Itoh 8510 SP. Remember, the proof is in the picture!

Requires 48K Apple II, II+, IIe, or Apple III in Emulation Mode. Order from your local Dealer. If they don't have it, we'll ship it to them within 24 hours.

Comparisons as of 3/1/84, based on Zoom Grafix (Second Edition) and Printographer (Version 4.3). Pictures shown are actual size, single magnification, and were produced on an Epson MX80 Printer.

Roger Wagner™ PUBLISHING, INC.

Circle 42 on Reader Service Card

10761 Woodside Avenue • Suite E • P.O. Box 582 • Santee, CA 92071 • 619/562-3670

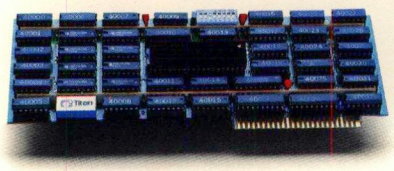
How to make your Apple run 3½ times faster.

Just plug in the Accelerator IIe™ and watch your Apple® II, II Plus or IIe take off.

Imagine running VisiCalc,® DB Master,® Applesoft, Apple Fortran or Pascal without long delays. With the Titan Accelerator IIe, your Apple II, II Plus or IIe runs these programs a true 3½ times faster. The Accelerator IIe also runs your Franklin, Basis and other Apple II compatibles 3½ times faster. This faster computer response time means less waiting for you and an increase in your productivity.

This Titan exclusive has its own fast 6502 processor, 64K of high-speed memory, and built-in fast language card. It is transparent to your software and is hardware compatible with most standard peripherals. And you can turn if off from the keyboard to run your Apple at regular speed.

Find out how to run faster and increase your productivity without breathing hard. For information on the Accelerator IIe and other Titan products, see your computer dealer or contact:



Titan TECHNOLOGIES, INC.
310 West Ann Street, Ann Arbor, MI 48104
Telephone (313) 662-8542

Apple is a registered trademark of Apple Computer Inc.
VisiCalc is a registered trademark of VisiCorp.
DB Master is a registered trademark of Stoneware, Inc.
Accelerator IIe is a trademark of Titan Technologies, Inc.

Sales and Marketing by The MARKETING RESOURCE GROUP, Fountain Valley, CA.

Circle 43 on Reader Service Card